2005

Comparison of Hyperspectral Imagery Target Detection Algorithm Chains

David C. Grimm

Comparison of Hyperspectral Imagery Target Detection Algorithm Chains

by

David C. Grimm

B.S. Electrical Engineering, Clarkson University, 2000

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
in the Chester F Carlson Center for Imaging Science
Rochester Institute of Technology

October 14, 2005

Signature of the Author David C. Grimm

Accepted by Illegible Signature
Coordinator, M.S. Degree Program Date

CHESTER F CARLSON CENTER FOR IMAGING SCIENCE
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. Degree Thesis of David C. Grimm
has been examined and approved by the
thesis committee as satisfactory for the
thesis required for the
M.S. degree in Imaging Science

John Schott
Dr. John Schott, thesis Advisor

John Kerekes
Dr. John Kerekes

David Messinger
Dr. David Messinger

Date

THESIS RELEASE PERMISSION

ROCHESTER INSTITUTE OF TECHNOLOGY

CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE

Title of Thesis:

Comparison of Hyperspectral Imagery Target Detection Algorithm Chains

I, David C. Grimm, hereby grant permission to Wallace Memorial Library of R.I.T. to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Signature David C. Grimm

Date

DISCLAIMER

The views expressed in this publication are those of the author and do not reflect the official policy or position of the U. S. Air Force, Department of Defense, or the U. S. Government.

Revised: 05-05-05

Comparison of Hyperspectral Imagery Target Detection Algorithm Chains

by

David C. Grimm

Submitted to the
Chester F. Carlson Center for Imaging Science
in partial fulfillment of the requirements
for the Master of Science Degree
at the Rochester Institute of Technology

Abstract

Detection of a known target in an image has several different approaches. The complexity and number of steps involved in the target detection process makes a comparison of the different possible algorithm chains desirable. Of the different steps involved, some have a more significant impact than others on the final result the ability to find a target in an image. These more important steps often include atmospheric compensation, noise and dimensionality reduction, background characterization, and detection (matched filtering for this research). A brief overview of the algorithms to be compared for each step will be presented.

This research seeks to identify the most effective set of algorithms for detecting a known target. Several different algorithms for each step will be presented, to include ELM, FLAASH, ACORN, MNF, PPI, N-FINDR, MAXD, and two matched filters that employ a structured background model OSP and ASD. The chains generated by these algorithms will be compared using the Forest Radiance I HYDICE data set. Finally, ROC curves and AFAR values are calculated for each algorithm chain and a comparison of them is presented. Detection rates at a CFAR are also compared. Since a relatively small number of algorithms were used for each step, there were no definitive results generated. However, a comprehensive comparison of the chains using the above mentioned algorithms is presented.

Contents

List of Figures

List of Tables

Missing Page

Chapter 1

Introduction

Detection of known substances, or targets, is a very common problem in hyperspectral imaging (HSI). Given a target with a known spectral signature, an algorithm that can decide if and where that target is present in an image is needed. In order to accomplish this, target detection algorithms such as matched filters have been developed to accentuate pixels in an image that contain the target.

HSI can be thought of as taking an image of the same scene with numerous (normally hundreds) different spectral bands. At each pixel, a spectrum measured represents the optical energy versus wavelength. These spectra are pictorially displayed as a third dimension in Figure 1.1. Each layer in the cube represents a spectral band; every band contains similar spatial information.

There are several unique steps to any target detection algorithm chain. Each of these steps plays a role in determining the overall performance of the target detector. Figure 1.2 is a flow chart detailing each of the more significant steps for a generic target detection algorithm chain. For each individual step along the way, there are several algorithms that will return an acceptable solution. It is desired to determine the *best* algorithm to accomplish each step. A particular combination of algorithms may work well for one image or target but may also perform terribly for a different image or target. Finding an optimal

combination of algorithms or recipe becomes an intriguing dilemma. In order to best accomplish this, commonly used algorithms for each step must be looked at and compared to each other for many different types of targets in an image. Each of the steps in the chain performs a distinct and important role in the overall target detection process. Known target spectra are in units of reflectance, since that is what is able to be measured in a lab or in the field. The data collected remotely by a sensor is in units of radiance. The difference between these measurements is the atmosphere between the remote sensor and the target. If the atmosphere was able to be effectively "removed", the collected information would be in the same units as the measured information. This is what the atmospheric compensation step seeks to accomplish.

Hyperspectral sensors generally collect hundreds of bands worth of information. This amount of data can be very computationally cumbersome or even impossible to work with. The dimensionality reduction step seeks to decrease the amount of data on which to operate without removing any pertinent information. Noise is a very common problem with any sensor. Noise can be removed simultaneously with other unnecessary information. Because of this, the noise and dimensionality reduction steps are often performed together.

One way to describe the background of an image is structured, or geometrically. That

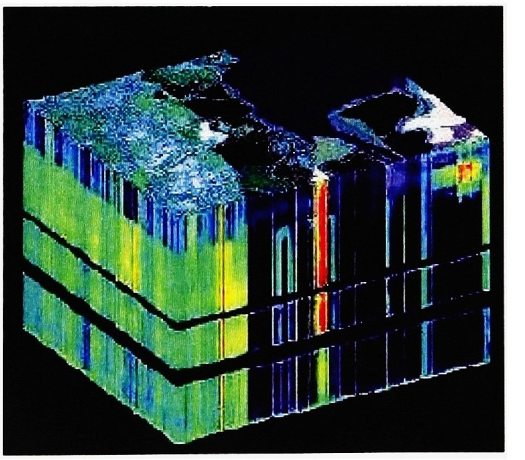

Figure 1.1: Example of an HSI image

is to say that a set of basis vectors is used to describe the background. These basis vectors are referred to as endmembers. The endmember selection step determines that set of basis vectors. Finally, the matched filters are the mathematical operations that determine the likelihood that a specific pixel contains a known target spectrum.

The following chapters discuss, in detail, the background and theoretical explanation of the algorithms studied in this effort. Results are presented in the form of ROC curves, average false alarm rates (AFAR) (Bajorski et al., 2004), and detection rates at a constant false alarm rate (CFAR). These metrics are used to compare several unique algorithm chains.

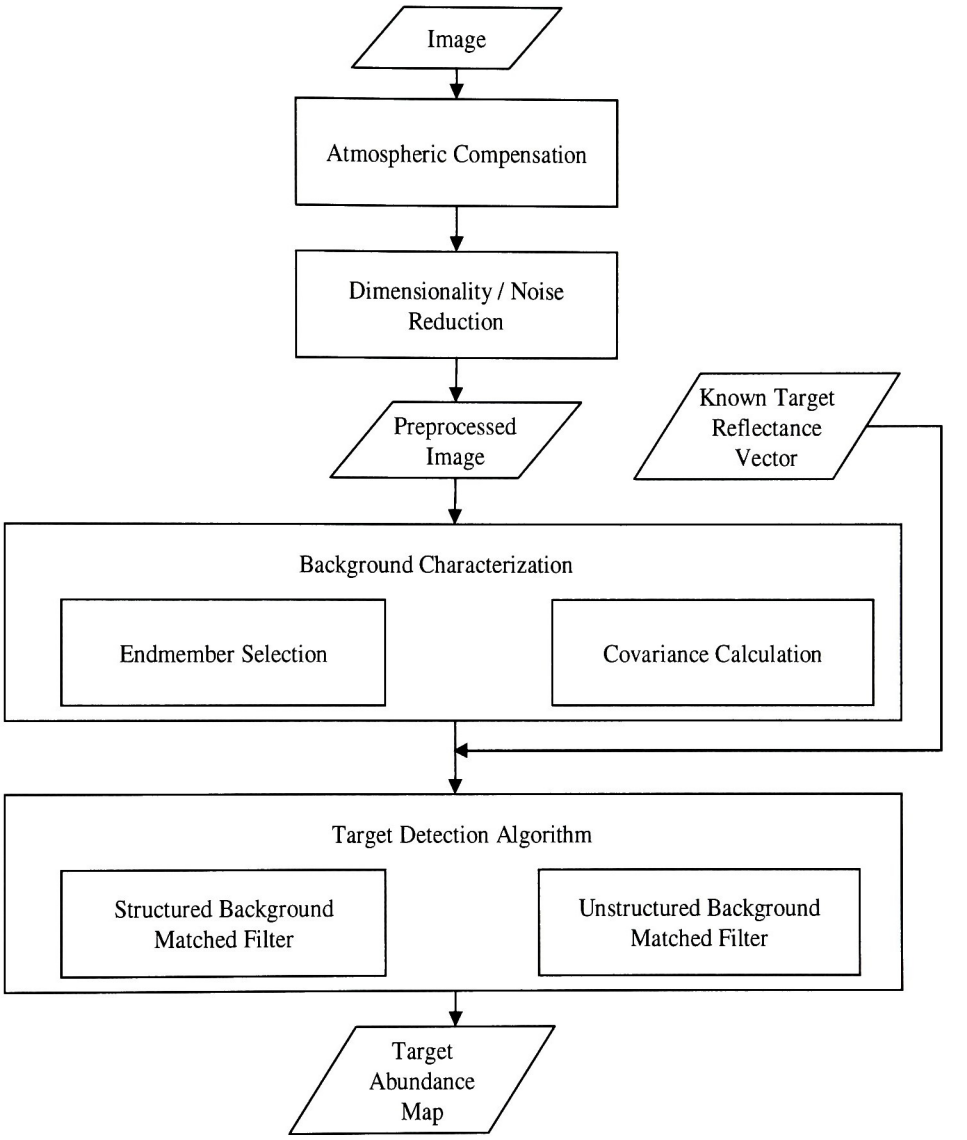

Figure 1.2: Typical target detection algorithm chain flow chart

Chapter 2

Atmospheric Compensation

Atmospheric compensation is the process of "removing" the illumination and atmosphere from the image. Often, the known spectral signature of a target consists of reflectance values measured in a lab. These measurements cannot take into account the effects of looking through thousands of feet of atmosphere. The ultimate goal of the atmospheric compensation process is to retrieve surface reflectance values from the radiance values recorded by the sensor. Equation 2.1 is one version of the sensor reaching radiance equation (ignoring thermal radiance) presented by Schott (Schott, 1997):

$$L_\lambda = [E'_{s\lambda} \cos\sigma' \tau_1(\lambda) \frac{r(\lambda)}{\pi} + F E_{ds\lambda} \frac{r_d(\lambda)}{\pi} + (1-F) L_{bs\lambda} r_d(\lambda)] \tau_2 + L_{us\lambda} \qquad (2.1)$$

where $E'_{s\lambda}$ is the top of atmosphere solar irradiance, σ' is the solar declination angle from the target centered z axis, τ_1 is the atmospheric transmission of the path from the sun to the target, r is the reflectance, F is the fraction of the hemisphere above the target that is unobstructed (sky), E_{ds} is the downwelled irradiance, r_d is the diffuse reflectance, L_{bs} is the reflected background radiance, τ_2 is the atmospheric transmission of the path from the target to the sensor, L_{us} is the upwelled radiance, and λ denotes the wavelength dependency of the associated terms. The terms in Equation 2.1 that the atmosphere directly contributes to are the two path transmittance terms, τ_1 and τ_2. As can be seen, these two terms - τ_2

in particular - will have a significant effect on the overall sensor reaching radiance.

All of the factors that determine the atmosphere - water vapor, molecular constituents, atmospheric density, etc. can substantially alter the way the same target looks under different conditions. As a result, the atmosphere through which an image was taken must be determined and taken into account. The simplest form of atmospheric compensation is the Empirical Line Method (ELM). There are several other algorithms that make use of atmospheric inversion principles: the Fast Line-of-Sight Atmospheric Analysis of Spectral Hypercubes (FLAASH) and Atmospheric CORrection Now (ACORN) are two of the more widely used atmospheric inversion based algorithms. These three methods are used in this work and will be discussed here.

2.1 Empirical Line Method (ELM)

ELM relies on ground truth inasmuch as at least two different regions in the image (preferably one dark and one light across each wavelength) have a known reflectivity. These regions must be at least one fully resolved pixel large. In order to ensure that an uncontaminated, fully resolved pixel is available, these regions generally have to be several pixels large in each direction. The regions must have corresponding ground truth reflectance spectra, ideally taken at the same time as the image. If such regions are not present in the image, an educated guess can be made by selection of regions for which an approximate reflectance can be created. For example, a white cloud may have a reflectivity of about 90% across all wavelengths. Obviously, there is a great opportunity for error introduction if estimates have to be made, but that can be limited if they are made smartly.

Once the two (or more) regions have been selected, a line is fit through the radiance (or digital counts) vs reflectivity points in each band. ELM assumes a linear relationship between the radiance (or digital counts) and the reflectivity. Mathematically, this relationship is expressed Equation 2.2,

$$L(\lambda) = m(\lambda) * r(\lambda) + b(\lambda) \tag{2.2}$$

where L is the observed radiance (or digital count), m is the slope of the line through the ground truth points, r is the reflectance, and b is the radiance (or digital count) value that represents zero reflectance. All of the terms have a wavelength dependency, λ. Equation 2.2 is a very simplified linear version of Equation 2.1. Figure 2.1 illustrates this linear relationship graphically.

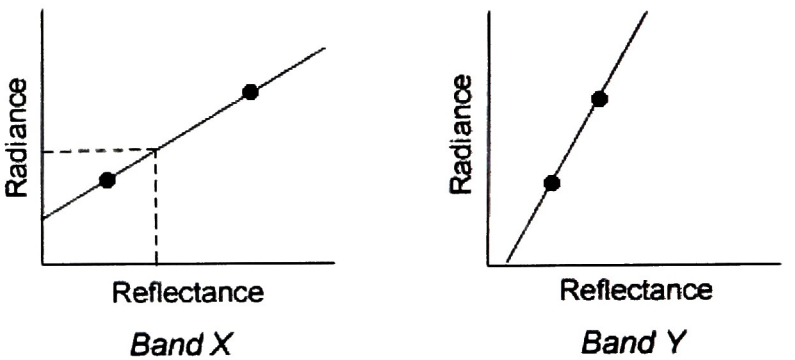

Figure 2.1: Graphical representation of ELM for two bands. The marked points are the "ground truth" points.

Now, the observed radiance in every pixel in the image can be converted into reflectance using the linear regression coefficients (m and b) obtained via Equation 2.2. It is important to note that ELM can work with both radiance or digital counts without calibration.

2.2 Atmospheric Inversion

An alternative method to ELM for calculation of atmospheric effects on an image is to characterize the atmosphere in an image to make "removal" possible. There are three main factors that, once determined, can adequately characterize the atmosphere: optical depth of the different aerosol and molecular constituents, surface pressure elevation, and atmospheric water vapor (Green et al., 1993).

Atmospheric inversion exploits the unique absorption features of the varying atmospheric materials at different wavelengths in order to arrive at the optical depth for the various aerosols present in the atmosphere. Figure 2.2 shows the absorption spectra for some common atmospheric molecular constituents. The strength, or depth, of these unique features is used to determine the quantity of the specific molecular constituent in the atmosphere.

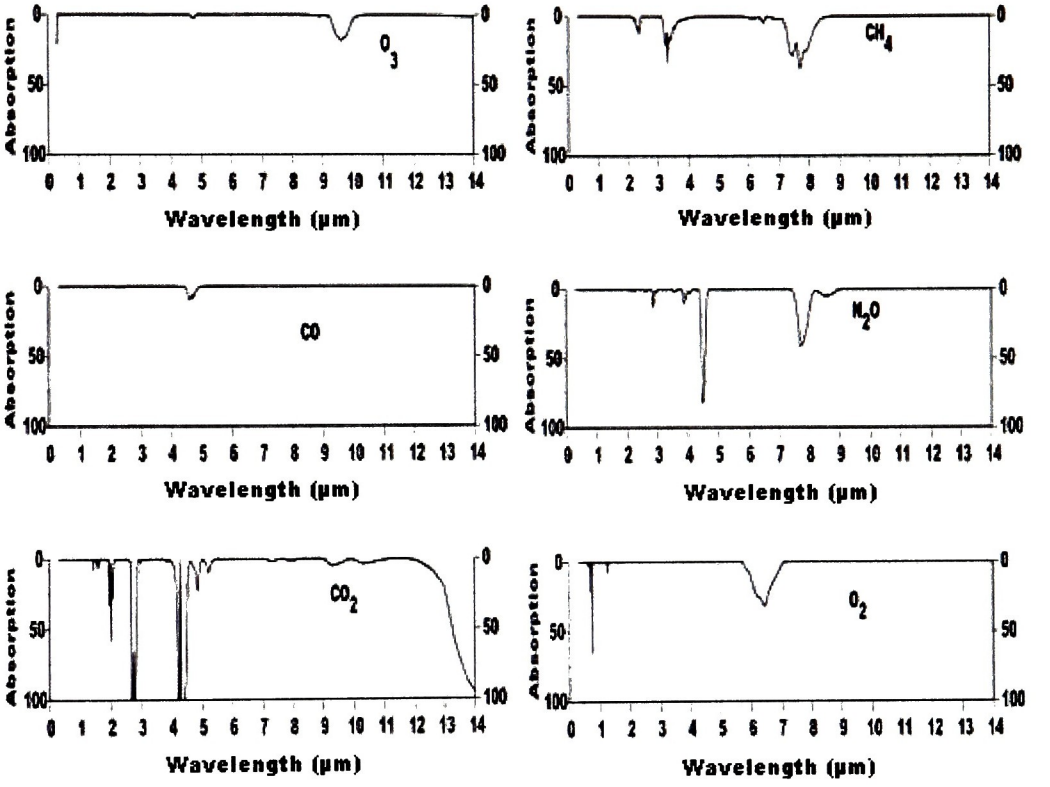

Figure 2.2: Absorption spectra of several common molecular constituents in the atmosphere (Schott, 1997).

The second factor that needs to be determined is the surface pressure elevation. This allows for compensation of atmospheric absorption due to well mixed gases in the atmosphere and the effects of molecular scattering (Green et al., 1993). The amount of oxygen is a good indicator of pressure depth. Oxygen can be treated the same as any other atmo-

spheric molecular constituent, and so the process for calculating the quantity of oxygen is the same as for the other molecular constituents. The absorption feature of oxygen present at $\approx$ 760nm is most commonly used in this process.

The third, and perhaps most important, contributor to atmospheric characterization is water vapor. The absorption spectrum for water is given in Figure 2.3. The absorption feature present at $\approx$ 940nm is commonly used to determine the amount of water vapor present in the atmosphere. There are also strong features at $\approx$ 1120nm and $\approx$ 1340nm that could also be used.

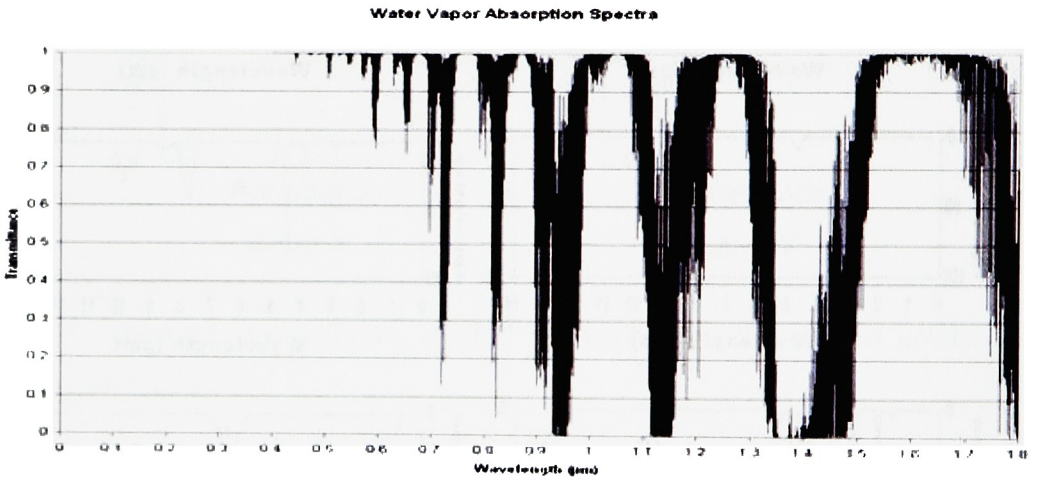

Figure 2.3: Absorption spectrum of water.

Simulation is used heavily in this process. MODTRAN (Berk et al., 2005) is commonly utilized to model the absorption features for many different concentrations of the varying aerosols in the atmosphere. These simulated absorption features are compared to the observed feature and the closest match determines the concentration of the aerosols present in the atmosphere of the given image. The non-linear least squared spectral fit (NLLSSF) algorithm (Green et al., 1993) is the primary determining metric for matching the modeled and observed spectra.

2.2.1 Fast Line-of-Sight Atmospheric Analysis of Spectral Hypercubes (FLAASH)

FLAASH is a commercially available atmospheric compensation algorithm developed by the Air Force Research Laboratory, Space Vehicles Directorate (AFRL/VS) to support the analysis of visible to shortwave infrared (0.4 μm to 2.4μm) hyperspectral and multispectral sensors (Berk et al., 2002). Using the atmospheric inversion method described above, FLAASH is able to determine the aerosol and water vapor content of the atmosphere in an image as well as the surface pressure. FLAASH was designed as a general purpose code developed in parallel with upgrades to MODTRAN. Originally, FLAASH was designed to work with MODTRAN4. Because of this "side-by-side" development and upgrade, FLAASH is able to exploit the speed gains and accuracy improvements obtained by ongoing improvements to MODTRAN. Another feature present in FLAASH, which will not be discussed in detail, is the ability to effectively remove clouds from an image. This is done by identifying cloud pixels in the image and replacing them with average radiance values of the pixels surrounding the cloud.

Equation 2.3 (Berk et al., 2002) is the sensor reaching radiance (L^*) equation utilized by FLAASH that has a different form than Equation 2.1 but will return the same value:

$$L^* = \frac{A\rho}{1 - \rho_e S} + \frac{B\rho_e}{1 - \rho_e S} + L_a^* \tag{2.3}$$

where ρ is the pixel surface reflectance (equivalent to r in Equation 2.1), ρ_e is an average surface reflectance for the area surrounding the pixel in question, S is the spherical albedo of the atmosphere (accounting for the skylight photons), L_a^* is the upwelled radiance, A and B are surface independent coefficients that vary based on the atmospheric and geometric conditions. All of the variables are implicitly wavelength dependent. The ρ_e is included to allow FLAASH to take into account the adjacency effects of the surrounding pixels on the pixel in question.

The per-pixel A, B, S, and L_a^* variables are all determined empirically via MODTRAN.

Once these values have been extracted, Equation 2.4 is used to solve for ρ_e (Berk et al., 2002):

$$L_e^* = \frac{(A+B)\rho_e}{1-\rho_e S} + L_a^* \tag{2.4}$$

where L_e^* is a spatially averaged sensor reaching radiance value.

Once ρ_e has been determined, Equation 2.3 can be solved for reflectance (ρ).

2.2.2 Atmospheric CORrection Now (ACORN)

Like FLAASH, ACORN is an atmospheric compensation algorithm that can model aerosol absorption in the atmosphere based on look-up-tables that are produced using MODTRAN4. However, this algorithm is capable of determining both molecular and aerosol scattering effects as well. The equation utilized by ACORN to represent the sensor reaching radiance is Equation 2.5 (ImSpec, 2004) :

$$L_t(\lambda) = \frac{F_0(\lambda)}{\pi}\left[\rho_a(\lambda) + \frac{T_d(\lambda)\rho(\lambda)T_u(\lambda)}{(1-s(\lambda)\rho(\lambda))}\right] \tag{2.5}$$

where L_t is the total sensor reaching radiance, F_0 is the above atmosphere solar irradiance, T_d and T_u are the downward and upward transmittance of the atmosphere, ρ_a and s are the upward and downward atmospheric reflectance, ρ is the surface spectral reflectance and λ denotes the wavelength dependency of the associated terms.

ACORN seeks to solve Equation 2.5 for the surface reflectance value of each pixel, ρ. This solution is given in Equation 2.6 (ImSpec, 2004) :

$$\rho(\lambda) = \left[\frac{\frac{F_0(\lambda)T_d(\lambda)T_u(\lambda)}{\pi}}{L_t(\lambda) - \frac{F_0(\lambda)\rho_a(\lambda)}{\pi}} + s(\lambda)\right]^{-1}. \tag{2.6}$$

A key unique feature of the ACORN algorithm is full spectral fitting to solve for the overlap of absorption due to water vapor and liquid water in surface vegetation (Kruse, 2004). The outputs of this software package are a water vapor image and a scaled surface reflectance cube. ACORN is commercially available and is distributed by ImSpec LLC.

2.3 Discussion

In this chapter, three different, valid atmospheric compensation techniques are presented. All three accomplish the same task, but there are inherent differences in the output of each algorithm. There are some similarities between the way FLAASH accomplishes this and the way ACORN does, but they are still quite different from each other. The major differences are the assumptions each algorithm makes and the complexity of its mathematical operations. ACORN is essentially a simpler version of FLAASH. FLAASH takes into account the adjacency effects of neighboring pixels in its calculation of surface reflectance, while ACORN does not have this feature. This alone will cause the outputs to be radically different from each other. The governing equations for FLAASH (Equations 2.3 and 2.4) and ACORN (Equations 2.5 and 2.6) are obviously not the same. Again, this introduces differences in the outputs of the algorithms. Finally, both FLAASH and ACORN differ greatly from ELM which assumes a linear relationship between the radiance values collected and surface reflectance.

Of all the steps looked at in this research, the atmospheric compensation step should have the greatest effect on the overall target detection performance. This is expected because this step actually changes the values present in the image. Also, this is the first step in the chain, so any errors introduced in this step will be amplified and can potentially cause errors in each of the subsequent links in the algorithm chain.

Chapter 3

Dimensionality/Noise Reduction

The next steps in the target detection process are dimensionality reduction and noise reduction. Since these two topics are closely related, the background for them both can be discussed simultaneously. Many algorithms tackle both problems at the same time.

The need for noise reduction is fairly obvious; as with any image processing, noise interferes with the actual signal. It is because of this interference that the noise should be reduced (ideally eliminated) before any processing is done on an image. The need for dimensionality reduction may not be as intuitive as noise reduction. As mentioned in Chapter 1, HSI relies on up to hundreds of spectral bands of information. Using all of the information contained in all of these bands can lead to dauntingly large data sets that may be computationally difficult or impossible to manage. Images will contain varying amounts of pertinent information in some bands while carrying little or no significant information in others. Also, it is likely that there is some information that is redundant across several bands and therefore can be reduced by discarding the repeated instances. This can be exploited, and the bands containing little amounts of data can be discarded reducing the amount of information to be operated on to a manageable level.

3.1 Principal Components Analysis (PCA)

Perhaps the most common method of dimensionality reduction is Principal Components Analysis (PCA). A common assumption associated with PCA is that variability equals information. Because of this, bands with high variance will dominate the first few principal components. Conversely, bands with low variance that may still contain useful information will be mixed in with noise from the more varying bands.

As is implied by the name, PCA breaks down the data into its principal components. In order to do this, the covariance matrix for the data must be constructed. The covariance matrix describes the dependant variability of the bands with relation to each other. Once the covariance matrix has been determined, its eigenvectors can be calculated. These orthogonal eigenvectors are used to rotate the original data into a new space that exactly de-correlates the data.

For the most part, the first few principal component bands contain the majority of the variability in an image. Because of this, PCA can be used to greatly reduce the number of dimensions on which to operate by disregarding the lower PCA bands. Figure 3.1 shows a simple example of PCA performed on a six band scene of Rochester, NY where the six PCA bands are displayed. There is no "right" way of determining the number of PCA bands to keep for an image. The amount of variation maintained must be weighed against the computational complexity of keeping a number of PCA bands. For the example in Figure 3.1, if only the first four bands were kept, 99.33% of the variability is maintained. This is enough to effectively reduce the dimensionality without greatly altering the amount of information on which to operate.

3.2 Maximum Noise Fraction (MNF)

A second algorithm that reduces dimensionality, the Maximum Noise Fraction (MNF) (Green et al., 1988), also works to improve the interaction between the noise and the signal.

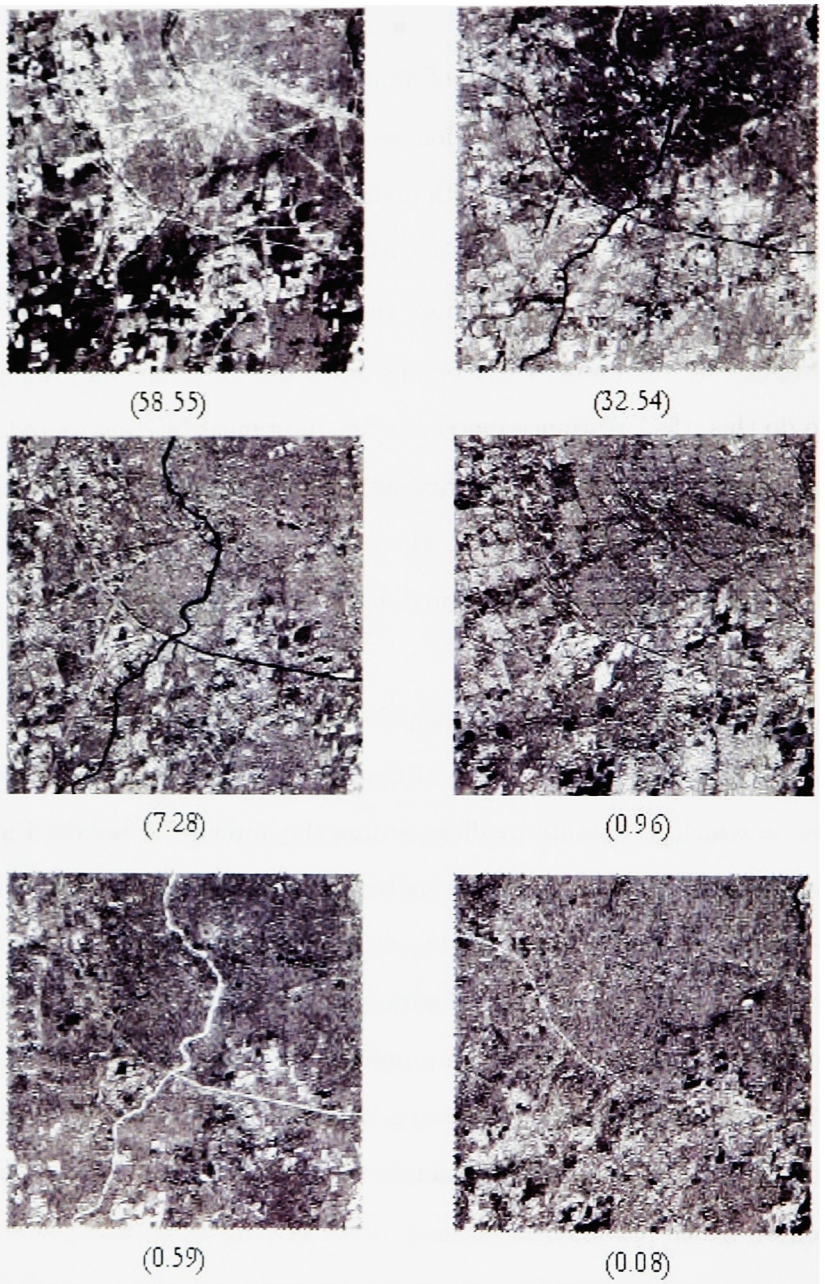

Figure 3.1: Principal component images from a six band image of Rochester, NY. Percentage of variance in each PC band in parentheses (Schott, 1997).

Essentially, the MNF algorithm is the same as the PCA process with some "pre-processing" steps added. For the MNF transform, the covariance matrix of the noise present is needed. One way to obtain this in-scene (if it is not known) is by locating a radiometrically flat field. Once a flat field is identified within an image, the mean can be subtracted from the observed values of that field. The result is a representation of the noise present in the image. Once this "noise image" is found, its covariance matrix can be calculated.

The next step in this process is basically to perform PCA on the noise image to de-correlate the noise. This is done by calculating the eigenvectors of the noise image. By definition, the resulting eigenvectors are orthogonal to each other. The original image can now be "passed through" this orthogonal transformation. After this is done, the orthogonally transformed image can be whitened, or normalized, by the eigenvalues of the noise covariance matrix. The final result of all these steps is an image in which all of the noise present is orthogonal and identical. Now, the PCA previously described can be performed.

The advantage of the MNF transform is that the "pre-processing" steps force the assumption used in PCA of variance representing information to be true. There is now equal variance present due to noise in the image. The bands are rank ordered by the signal to noise ratios and the highest PC bands are retained. The result is a greatly reduced dimension space with little noise to interfere with target detection.

3.3 Discussion

As described above, the process involved in both PCA and MNF is very similar, however, the results will be dramatically different. This difference stems from the fact that MNF uses the eigenvectors from a different matrix to perform the rotation in an effort to reduce noise. Calculation of the eigenvectors and the actual rotations are rather trivial mathematical operations. Neither PCA nor MNF change the amount of information present in the data. However, in order to reduce dimensionality, some of the lower components have to be thrown out. This will reduce the amount of information retained, by design. If the information

removed is noise (as is the goal of MNF), the overall performance of the algorithm chain can improve. Conversely, if the removed PC bands contain some pertinent data, the overall performance of the algorithm chain can be greatly degraded.

If the "noise image" does not characterize the noise present in the image well, the eigenvectors derived will not rotate the data into a space that is orthogonal to the true noise in the image. What will end up happening is the data will be rotated into a space that is orthogonal to some of the pertinent information causing poor information to be passed on to the next step in the chain. Another situation that will hinder the ability of MNF to improve the overall algorithm chain, is there being very little noise in an image to begin with. Extremely low noise is very difficult to characterize well.

The algorithm for this step that is expected to most greatly improve the detection results is the "none" case. This is due to the fact that no information is being removed from the original image. By performing PCA or MNF, that sheer amount of data is reduced to a point where it is much more manageable computationally. Clearly, there is a trade off here, however this aspect will not be discussed in detail in this research. The focus is solely on which algorithm will contribute to the best overall detection results.

Missing Page

Chapter 4

Endmember Selection

There are two overarching ways to characterize the background of an image: structured or unstructured. A structured background is one that is defined by a set of basis vectors. This set of basis vectors, or endmembers, completely defines every non-target pixel in an image, in theory. An unstructured background is defined statistically, normally by a covariance or correlation matrix. While a background independent matched filter, the Spectral Angle Mapper (SAM), and an unstructured matched filter, the generalized likelihood ratio test (GLRT), were used in determining the targets, this research focuses on matched filters that employ a structured background.

All targets fall into two very broad categories: resolved and unresolved. A fully resolved target is one that fills at least one entire pixel in an image. An unresolved target is one that does not fully fill a pixel. For targets that are unresolved, it becomes necessary to determine the constituents of each pixel when using a structured background model. In order to accomplish this, each pixel must be represented as a combination of its components (or endmembers). A common representation, known as the linear mixing model (LMM), for a mixed pixel is shown in Equation 4.1:

$$\vec{L} = \vec{E}\vec{F} \tag{4.1}$$

where L is the radiance vector of the pixel, E is the endmember matrix, and F is the vector of unknown endmember fractions. This representation is based on the assumption that each pixel is a linear combination of the image endmembers. Every pixel in an image, whether it contains target or not, can be represented using the LMM. However, the accuracy is determined by how well the set of endmembers characterizes the scene.

In order to enable the matched filters to properly suppress the background in an image, the background must first be well characterized. This is accomplished by deriving the endmembers used to define the background pixels. In theory, the background is fully characterized by a matrix consisting of *all* the endmembers that mix to define *all* non-target signals present in the image. This would represent an ideal situation that would result in endmember matrices that could be almost as large as the image itself and rendering the matched filters nearly too computationally cumbersome to use. In practice, several endmembers are used to represent the entire background. There are several different methods of eliciting these endmembers directly from an image, three will be discussed here: PPI, MAXD, and N-FINDR.

4.1 Pixel Purity Index (PPI)

The first endmember selection algorithm is the Pixel Purity Index (PPI) method (Boardman et al., 1995). Using PPI, all of the pixel values in an image are projected onto randomly selected vectors. Each time a projection is accomplished, the extreme points are noted. Spectrally pure pixels will consistently be extreme points of these random projections. A major assumption of this method is that spectrally pure pixels (which are endmembers) are present in the image. This is generally a good assumption. This method is also rather susceptible to noise, so steps (*e.g.* thresholding) should be taken to help lessen the negative effects of noisy pixels. Figure 4.1 is an example projection of data points.

4.2 MAXD

The second endmember selection technique is MAXD (Lee, 2004). Similar to PPI, MAXD also uses projection to help sort out the endmembers of an image. Given a set of data points, MAXD begins by selecting the points that have the maximum and minimum Euclidean distance from the origin. These two points are the first two endmembers. Once these points are selected, all of the data points are projected onto a vector orthogonal to the difference vector between the first two, and the pixel the maximum distance from the common point projection of the first two is selected as the next endmember. This process is repeated until the desired number of endmembers is found. Figure 4.2 pictorially explains how this process works.

4.3 N-FINDR

Another method of determining endmembers in an image is N-FINDR (Winter, 1999). This method again looks for extrema points in the data set. The set of data points for an image in N-dimensional space can be thought of as a convex-hull, where one more

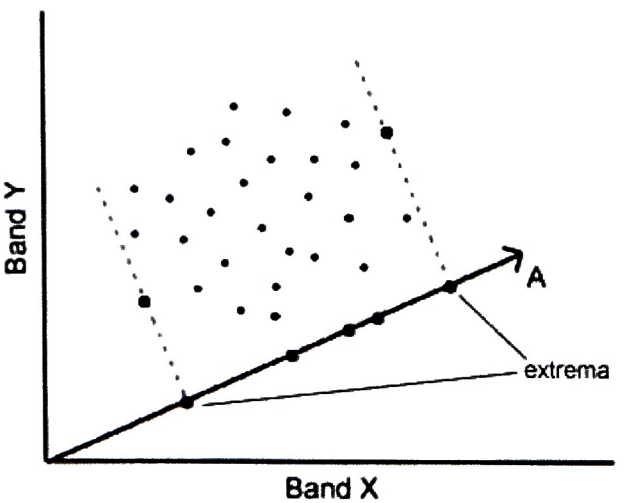

Figure 4.1: Example PPI projection (Boardman et al., 1995).

endmember than dimensions is needed to make a closed convex hull. For example, given an image with two bands (dimensions), three endmembers form a triangle as a closed convex hull (Figure 4.3). This algorithm was originally packaged for use in unmixing pixels in an image as opposed to target detection. However, both processes require endmembers to be elicited from the image first. Because of this overlap, N-FINDR is able to be used for background characterization due to its ability to return the spectra of all the endmembers it uses to unmix an image.

N-FINDR assumes that spectrally pure pixels can represent endmembers. Clusters of

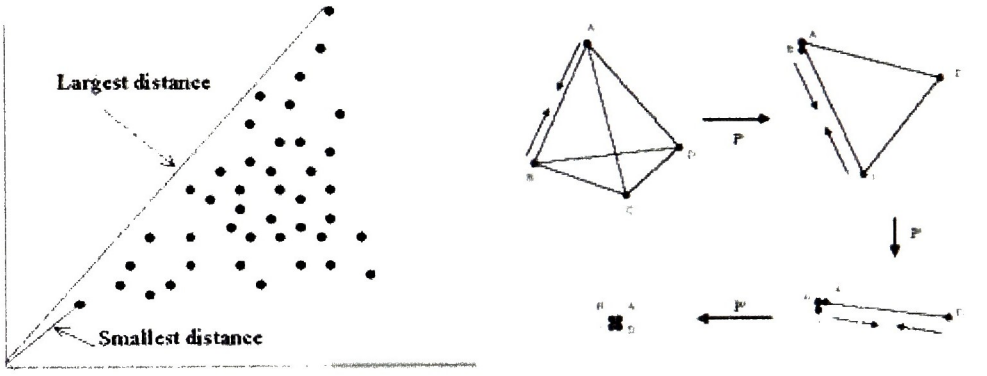

Figure 4.2: MAXD endmember selection algorithm (Lee, 2004).

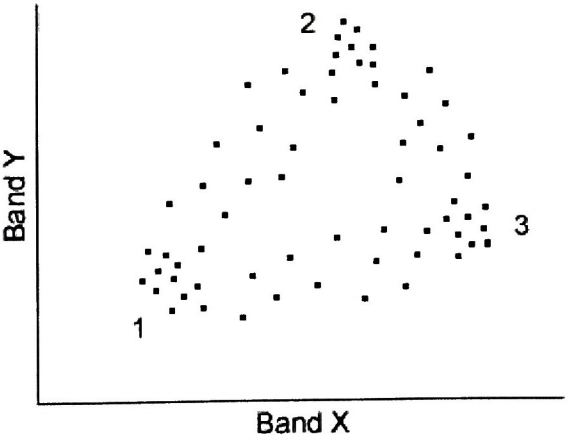

Figure 4.3: Convex hull for a two band image.

pixels around the area of a "point" on the convex hull, such as the clusters labeled 1, 2, and 3 in Figure 4.3, are indications of spectrally pure pixels. Spectrally mixed pixels generally appear between the pure pixels, sometimes outside of the convex hull, and are recognized as non-endmembers by N-FINDR.

As a starting point, N-FINDR randomly selects a group of pixels (automatically determined by a statistical analysis of the image (Senna Consulting, 2004)) so that a convex hull can be formed. Once these points have been chosen, the volume of the convex hull is computed. This is repeated until every possible combination of points that could form a convex hull has been used and has a volume associated with it. The set of points resulting in the largest volume is then selected as the end members. Figure 4.4 graphically shows the maximum volume convex hull selected for a two band image.

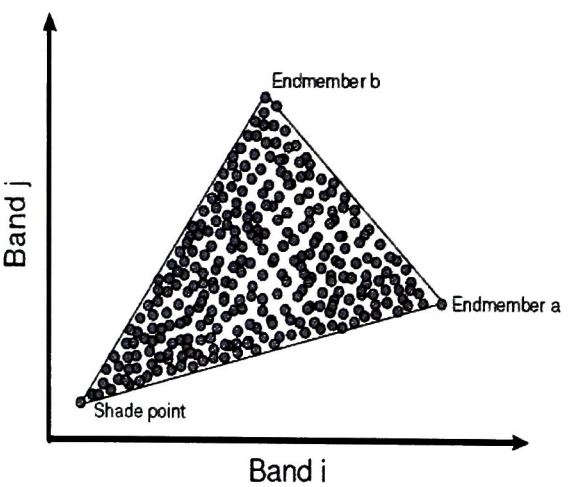

Figure 4.4: Maximum volume convex hull for a two band image (Senna Consulting, 2004).

This algorithm is also very sensitive to noise. So, as with PPI and MAXD, steps need to be taken to decrease the unwanted effects of noisy data.

4.4 Discussion

The ultimate goal of this step in the algorithm chain is to characterize the background of an image by determining a set a basis vectors representative of the background. All three of the algorithms discussed in this chapter can satisfactorily accomplish this goal. However, all three will return a different set of basis vectors. This is because all three of the algorithms are very different from each other in their implementation. The two that are most similar are MAXD and PPI. Both of these project the data set onto vectors and record the extrema points. The major difference in theory is that MAXD will use pixels taken directly from the image and PPI is actually an average of several pixels from the image. In practice, MAXD is fully automated, but PPI requires a great amount of user input. Chapter 6 discusses this reliance on the user by PPI in greater detail.

N-FINDR uses an entirely different technique to determine the basis vector set. This difference alone would result in different vectors being elicited. As with PPI, N-FINDR also averages several similar pixels to arrive at each basis vector. This option exists to help mitigate the inherent susceptibility to noise of N-FINDR, but can be skipped causing the vectors chosen by N-FINDR to be pulled directly from the image. All three of these techniques are viable options for selecting endmembers. But, for some cases, one may (or may not) be a better option.

Chapter 5

Matched Filters

One of the most common ways to go about searching an image for a known target is by use of matched filters. Basically, a matched filter scans an image looking for a specified spectral signature. It accentuates pixels that "look" similar to the target. All matched filter approaches rely on the same basic hypothesis test: H_1 target present, or H_0 - target absent. Each pixel in an image is interrogated and returns an answer proportional to the likelihood of the target being present in that pixel. The responses to the matched filters are then thresholded to return an answer to this binary hypothesis test. In order to accomplish this test, the matched filter must attempt to reject the background (any signal present that is not from the target) and amplify the target. The matched filter used in any given circumstance is based on the type and amount of *a priori* knowledge of the target, the background, the noise present and their mutual interaction.

All matched filters can be separated into three basic categories defined by the way the image background is modeled: 1) background independent, 2) structured or geometric background models, and 3) unstructured or stochastic background models. As noted in Chapter 1, SAM is a background independent matched filter and is the only one of this variety discussed. The GLRT is the only stochastic matched filter employed in this research as the focus is on structured background model matched filters.

5.1 Matched Filters Using Structured or Geometric Background Models

Structured background models rely on the assumption that every pixel in the image can be represented as a combination of the constituent materials in a scene. Commonly, this mixture is thought to be linear as discussed in Chapter 4. There is a plethora of ways to determine the endmembers mixing to create the scene pixels, several of which are described in Chapter 4, and it is important to note the dependency these matched filters have on that process. For structured background models, the two competing hypotheses can be represented by Equations 5.1 and 5.2.

$$H_0 \quad : \quad \vec{x} = \mathbf{U}\vec{\alpha_b} + \vec{w} \tag{5.1}$$

$$H_1 \quad : \quad \vec{x} = \vec{d}\,\alpha_t + \mathbf{U}\vec{\alpha_b} + \vec{w} \tag{5.2}$$

where $\vec{x}$ is the (L-band x 1) pixel of interest, $\mathbf{U}$ is the (L x Q) matrix having Q background vectors, $\vec{\alpha_b}$ is the (Q x 1) relative abundance of the background vectors $\mathbf{U}$, $\vec{w}$ is the (L x 1) noise vector, $\vec{d}$ is the (L x 1) target vector and α_t is the (scalar) target abundance.

There are four fundamental assumptions for target detection using the structured background model:

1. Every pixel can be represented by the linear mixing model
2. The modeling error is uncorrelated with a multivariate normal distribution
3. The background spectra are known
4. The target spectra are known

All of the structured background algorithms attempt to accomplish the same end: project the entire image into a space orthogonal to the background space. When the background space is properly defined, the orthogonal projection effectively removes the

background from the image and gives a measure of how "target-like" each pixel is. Figure 5.1 offers a graphical representation of the operations involved in the geometric background model algorithms.

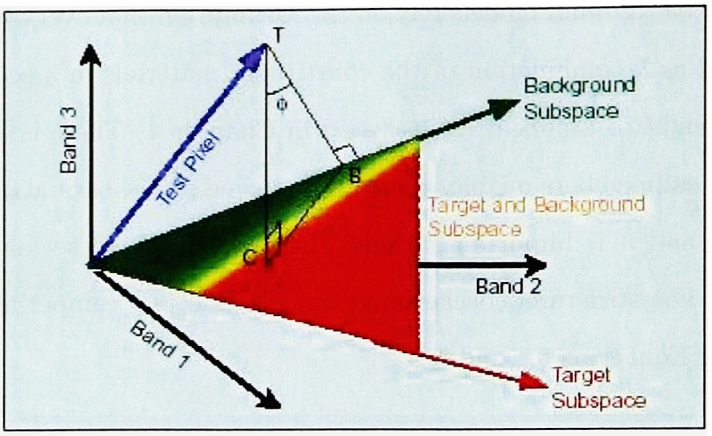

Figure 5.1: Graphic representation of structured background model matched filter operations (Manolakis and Shaw, 2002).

The ultimate goal of the target detection algorithm chain is *target detection.* The means to accomplish target detection in this case are matched filters. The two similar but different structured background matched filters compared by this research are the Orthogonal Subspace Projector (OSP) and the Adaptive Subspace Detector (ASD).

5.1.1 Orthogonal Subspace Projector (OSP)

The test statistic for the OSP detector is written as

$$T_{OSP}(x) = \frac{d^T P_b x}{d^T P_b d} \tag{5.3}$$

where d is the desired target vector and $P_b = I - U(U^T U)^{-1} U^T$ with I being the (bands x bands) identity matrix and U being the matrix of endmember vectors (Harsayni and Chang, 1994). In words, $P_b x$ is the projection of the pixel, x, onto the vector orthogonal to the background subspace.

OSP projects each pixel into a space orthogonal to the background of the image which is represented by the background endmembers elicited from the image. This, in theory, forces the target pixels to "stick out" from the background. The pixels that most closely match the given target spectra receive a high T_{OSP} value.

5.1.2 Adaptive Subspace Detector (ASD)

A second implementation of a structured background model detector is ASD. The test statistic for this detector is represented mathematically as

$$T_{ASD}(x) = \frac{x^T(P_b - P_S)x}{x^T P_S x} \tag{5.4}$$

where $P_S = I - S(S^T S)^{-1}S^T$ with $S \equiv [dU]$ the matrix formed by the concatenation of the target spectra and the endmember spectra, and the other variables are the same as those defined for Equation 5.3 (Manolakis et al., 2001).

Both OSP and ASD suppress the image background by projecting each interrogated pixel onto a space orthogonal to the background via the P_b operation. The major difference between OSP and ASD is the P_S term used in ASD. This is the projection of the pixel onto a space orthogonal to the background *and* the target. The difference between these two projections, P_b and P_S, is a measure of just how "target-like" the pixel is. This is the only difference between these two matched filters, but it is enough to cause the results from the two to be drastically different.

Chapter 6

Methodology

The previous chapters discussed the theory behind all of the algorithms used for this research. All of the implementations that were alluded to were general and did not go into too much detail on exactly how to go about using the various algorithms. This chapter will detail exactly how each algorithm was used, the inputs and outputs for each step, and grant an understanding of how the results presented in the following chapter were generated. Whenever possible, all algorithms were performed within the ENVI environment.

6.1 Data Set

The data set used for this research was an image from the "Forest Radiance I" experiment collected by the HYDICE (Rickard et al., 1993) sensor. The image dimensions are 320x1280 with 210 spectral bands covering the spectrum from approximately 350nm to 2500nm. A "border" of non-image pixels is present on either side of this image. These two "borders" were taken out of the image by removing ten columns of pixels from either side of the image. Of the 210 spectral bands, some were excluded, particularly covering the water and oxygen absorption bands, leaving 145 "good" bands. All calculations were performed on the resulting image of dimensions 300x1280 with 145 spectral bands.

Eight targets of varying degrees of difficulty were used to compare the algorithm chains.

There are many man-made targets placed into this scene including six calibration panels varying from 2% gray scale to 64% gray scale reflectors. In order to determine the overall difficultly of detection for each target, two basic matched filters were run for every target in the scene: SAM, and GLRT.

SAM is a background independent target detection algorithm that relies on the spectral angle between the target and pixel in question to determine if the target is present. The test statistic is computed as

$$T_{SAM}(x) = \frac{d^T x}{(d^T d)^{1/2}(x^T x)^{1/2}} \tag{6.1}$$

where d is the target spectrum and x is the pixel spectrum in question. For the purposes of this research, the SAM results were used in determining the targets to be compared and the difficulty of their detection.

The GLRT relies on an unstructured, or stochastic, representation of the image background in the form of a covariance matrix and, for this implementation, a pre-defined target spectrum. The GLRT is written as

$$T_{GLRT}(x) = \frac{(d^T \Sigma^{-1} x)^2}{(d^T \Sigma^{-1} d)(1 + x^T \Sigma^{-1} x)} \tag{6.2}$$

where d and x are the same as Equation 6.1 and Σ is the covariance matrix used. The GLRT results were generated using the covariance of a spatially selected region consisting of "tree" pixels in the image. Both SAM and GLRT operated on an ELM corrected image.

By simply looking at the image (Figure 6.1), it can be seen that, barring the planted targets, the scene is rather uncluttered with little man-made material. There are only about six or seven different classes of pixels present in the image and several of those are rather similar (*i.e.*, light and dark grass are considered unique classes). Secondly, there are no "concealed" target pixels, all of them are out in the open and completely uncovered. There are both fully resolved and sub-pixel target pixels for all target types.

6.2 Target Selection

Section 6.1 states that there are many planted man-made targets in the image. In actuality, there are 40 such unique materials deployed in the scene. To use all 40 targets for comparison would be prohibitive. So, choosing a smaller subset of targets that is representative of the larger group becomes necessary. The targets can be lumped into three main groups: panels and tarps, vehicles, and netting. So, at least one target from each of these groups was selected. Secondly, targets of varying sizes are present, so both large and small targets were chosen. The amount of pixels per selected target ranges from 9 to 347.

Lastly, some targets are easier to detect than others due in large part to the amount of contrast that exists between the target and scene background (*i.e.*, it is easier to find a needle in a haystack than a specific needle in a stack of needles). It is vitally important that targets of differing detection difficulty were chosen for comparison. In order to determine just how easy or difficult a particular target is to detect, some empirical method has to be devised. The two general detectors discussed in Section 6.1 were used for this purpose. SAM and GLRT were both used to detect each of the targets in the image with all of the other planted targets masked out so as to not provide false alarms.

Based on the above mentioned criteria, eight targets were selected. Table 6.1 below

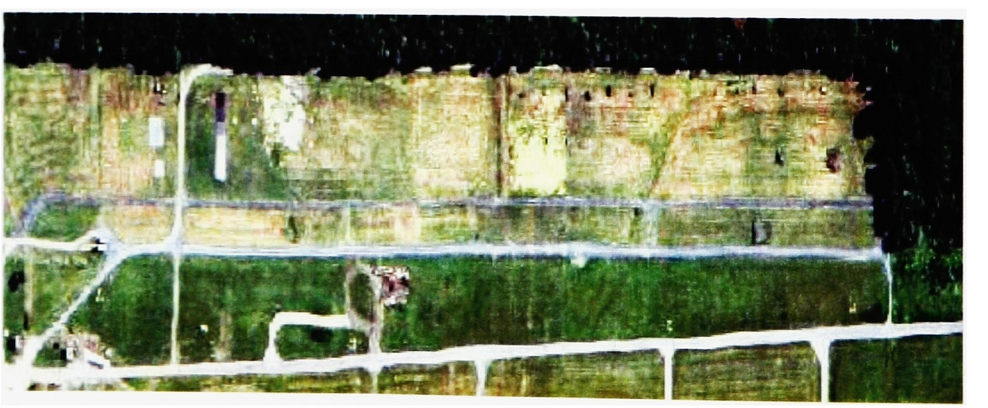

Figure 6.1: Three band RGB representation of the HYDICE sensor data set Run 05 from the Forest Radiance I experiment.

provides a list and description of the targets selected.

Name	# of Pixels	Target Contrast
C5	344	high
F3	35	high
F4	33	high
F8	9	low
F13	20	high
V1	67	low
VF1	50	low
VF6	44	low

Table 6.1: Table of selected targets.

6.3 Atmospheric Compensation

The Forest Radiance I data set came with an ELM and a FLAASH corrected image included in the distribution. The ACORN compensation had to be performed on the original image using the software package distributed by ImSpec LLC.

The ground truth points used for the ELM compensation were the calibration panels placed in the image. The panels were measured at the time the overhead imagery was taken and therefore are the best possible approximations for those particular panels. The resulting compensated image is comprised of floating point precision values in unscaled surface reflectance units.

FLAASH was run using an IDL interface with a combination of text menus and a graphical user interface (GUI). The IDL code has since been incorporated into the ENVI program and a FLAASH GUI is now available in ENVI. The output file from FLAASH is the reflectance scaled into 2-byte signed integers (RSI, 2004). The scale factor used for this data set is 10,000.

ACORN requires rather simple input parameters, primarily information about when and where the data were collected so as to decide which atmospheric parameters to use for the inversion. In order to properly perform the necessary calculations, the values of the input file must be in units of $W/m^2/\mu m/sr$ (ImSpec LLC, 2004). A scale factor of $\frac{1}{75}$ was applied to the calibrated HYDICE file in order to achieve the necessary radiance units for the ACORN program. Like, FLAASH, the values of the ACORN compensated output file are 2-byte signed integers that are the reflectance scaled by 10,000.

6.4 Dimensionality / Noise Reduction

Both of the algorithms used for this step of the algorithm chain were implemented via ENVI. PCA was performed directly on the atmospherically corrected images using the option available in ENVI. The goal of PCA is to reduce the dimensionality of the image to be operated on and keep as much variability as possible. Well over 99% of the overall variability of each image was preserved. Table 6.2 details the number of PCA bands retained for each image and the variability associated with those bands. The option to use the scene covariance matrix in the transform was selected.

Atmospheric Compensation Method	PCA Bands Retained	% of Variability
ELM	20	99.82
FLAASH	20	99.85
ACORN	20	99.96

Table 6.2: Number of bands and total percentage of variability retained after PCA was performed on the atmospherically compensated Forest Radiance I images

The similarities between PCA and MNF were outlined in Chapter 3. The major difference is the need for the statistics to characterize the noise in the image. ENVI can calculate the noise statistics of an image automatically by selecting the "Estimate Noise Statistics from Data" option in the MNF Transform menu. ENVI was not able to calculate noise

statistics from the ACORN compensated image. Due to the inherent similarities between FLAASH and ACORN, the noise statistics computed for the FLAASH compensated image were used to perform the MNF transform on the ACORN compensated image. Table 6.3 contains the number of MNF bands retained for each image and the variability associated with those bands.

Atmospheric Compensation Method	MNF Bands Retained	% of Variability
ELM	136	98.14
FLAASH	137	98.22
ACORN	50	99.86

Table 6.3: Number of bands and total percentage of variability retained after MNF was performed on the atmospherically compensated Forest Radiance I images

The case where no dimensionality/noise reduction was performed is also explored in this research. The statistic files generated by ENVI used to accomplish the forward PCA and MNF transforms were retained for use later to perform a reverse transform.

6.5 Background Endmember Selection

Of the three different endmember selection algorithms, only one, PPI, is able to be run in ENVI. The other two methods required other software in order to be run. The following sections will detail exactly what software was used and all of the image processing accomplished before running the different endmember selection algorithms.

6.5.1 PPI

The ENVI PPI implementation is not a single, stand-alone option and requires some user intervention. The first step is to input the number of iterations (random vector projections) that ENVI will perform. ENVI uses a threshold factor to help mitigate noise interference. The threshold factor determines the number of pixels other than strictly the extreme pixels

considered - pixels within the threshold factor of the extreme pixels are considered extrema as well. This must be input at the same time as the number of iterations. In this case, 30,000 iterations were run for all three images. For the ELM image, a threshold of 0.02 was used and for FLAASH and ACORN, a threshold of 2 was used. The difference in thresholds is due to the scaling of the FLAASH and ACORN images mentioned in Section 6.3. ENVI also allows for the use of a pixel mask. Masked pixels are effectively removed from the image prior to PPI being run. All of the targets planted in the image were masked out.

The next step is for the user to decide how many times a pixel must be detected to be considered a possible endmember. This is a judgement call on the part of the user. The number of apparent background classes was discussed in Section 6.1. So, the number of pixels looked at as endmembers was selected so as to allow for 6 to 10 different clusters.

Once the number of pixels to be considered is determined, they must be clustered. This is done using the n-dimension visualizer in ENVI. The significant pixels are clustered by the user. The mean of each cluster of pixels is then used as an endmember; the number of clusters found is equal to the number of endmembers elicited from the image.

Table 6.4 contains the least number of times a pixel was identified as extreme to be considered a possible endmember, the total number of pixels at or above that threshold, and the resulting number of endmembers for each image used.

6.5.2 MAXD

There is no implementation of MAXD in ENVI. Since MAXD was originally developed at Rochester Institute of Technology, an in-house IDL program that was written by Dr. Harvey Rhody was used. This IDL code is not currently commercially available. Unlike PPI, the MAXD code is a fully automated process where the only user input is the number of endmembers to be found. Past research has shown that there is an optimal number of endmembers that can be found via MAXD for a scene. Selection of either too many or too few has a negative impact on detection results using the background characterized by

MAXD (Bajorski and Ientilucci, 2004). When using MAXD on the Forest Radiance I data set, between 8 and 14 endmembers works well to characterize the scene. So, the user defined maximum number of endmembers to be found by MAXD was set at 10 for all images. As with PPI, all of the planted targets in the scene were masked out prior to running the MAXD routine.

6.5.3 N-FINDR

The software to run N-FINDR is available commercially and is distributed by ImSpec LLC. The version used for this research is N-FINDR 3.0 provided by Michael Winter. Like the other two endmember selection methods, the planted targets in the scene were removed prior to the application of the N-FINDR routine. However, unlike the others, this presented a bit of a dilemma. The dimensions of the image do not matter when running PPI or MAXD. Such is not the case with N-FINDR. The image must "look" like an image in order for the

Atmospheric Compensation / Endmember Selection Combination	**Pixel Detect Threshold**	**# of Pixels Selected**	**# of Endmembers Elicited**
ELM / None	15	546	7
ELM / PCA	200	1646	7
ELM / MNF	10	269	6
FLAASH / None	10	344	6
FLAASH / PCA	10	271	8
FLAASH / MNF	25	969	7
ACORN / None	10	405	6
ACORN / PCA	10	340	7
ACORN / MNF	200	1801	7

Table 6.4: The number of pixels considered and the number of endmembers selected using PPI for each image.

N-FINDR program to properly run. That is to say that the image must be in the format (X x Y x bands). The removal of the planted targets changes the dimensions of the image to be (pixels x bands) and removes several pixels from the image. The result is an image that is (pixels x bands) but cannot be returned to its original dimensions. Luckily, the resulting number of pixels in the image after the targets have been removed was such that they could be reoriented into (X x Y x bands) form. However, all spatial information is lost, but spectral integrity is retained and that is all that is important for this step. The size of the images fed to the N-FINDR program was (393 x 959 x 145).

The second difficulty with N-FINDR was encountered when running it on a PCA or MNF image. Unlike the other two methods, N-FINDR returns only the spectra of the selected endmembers, not location. So, when being run on a PCA or MNF image, the PCA or MNF spectra is returned. In order to combat this, the returned spectra are run through the reverse PCA or MNF transform using the saved statistics files mentioned in Section 6.4. The result is equivalent to the spectra of the selected pixels in their respective original atmospherically compensated image.

N-FINDR is able to statistically determine the "correct" number of endmembers to retrieve for a scene. It also allows the user to enter a maximum number of endmembers to find. For each N-FINDR run, the maximum number of endmembers was set sufficiently high to allow the program to determine the number on its own. Table 6.5 lists the number of endmembers the N-FINDR program chose to find for each image.

The second user input is the approximate signal to noise ratio (SNR) of the sensor. The SNR used for the HYDICE sensor was 150. Finally, in order to help alleviate the inherent susceptibility to noise of the algorithm, an option to "average endmembers to reduce noise" is offered and was selected. This option groups pixels found as endmembers that are spectrally similar and uses the mean of these groups as the representative endmember for those pixels.

6.6 Detection and Metrics

Neither of the two matched filters are available for use in ENVI. However, they are rather simple mathematical operations. A simple IDL code was written for each matched filter and was used to implement them.

6.6.1 ROC Curve and AFAR Generation

The actual detection statistics were calculated for every pixel in the image, including all of the planted targets. However, when the false alarm rates and detection rates were calculated, the results for the pixels that were planted materials and not the target in question were not considered. As a result, detection of these pixels was not treated as a false alarm.

ROC curves generally plot the probability of false alarm against the probability of detection. Since the exact location of each target pixel in the image was known, it was possible to estimate these probabilities as sample rates. The ROC curves generated are

Atmospheric Compensation / Endmember Selection Combination	**# of Endmembers Elicited**
ELM / None	6
ELM / PCA	6
ELM / MNF	30
FLAASH / None	8
FLAASH / PCA	9
FLAASH / MNF	27
ACORN / None	8
ACORN / PCA	10
ACORN / MNF	15

Table 6.5: The number of endmembers automatically selected by N-FINDR for each image.

false alarm rate (the actual fraction of false alarms encountered) versus detection rate (the actual fraction of target pixels encountered). The AFAR was computed by simply averaging each false alarm rate at each detection rate. AFAR can be used as a general guideline as to performance (lower is better), but for detailed comparison, the ROC curves provide better insight.

Missing Page

Chapter 7

Results

7.1 Atmospheric Compensation

Of the three atmospheric compensation algorithms discussed in Chapter 2, it is expected that ELM will produce the best results for this particular data set. This is due to the high level of ground truth with this data set. The accuracy of the line fit between ground truth points is increased greatly with a higher level of ground truth (*i.e.* calibration panels). This assertion is supported by Figures 7.1 and 7.2 which show the ELM compensated pixels fitting much closer to the measured reflectance values than either the FLAASH or ACORN compensated pixels. It should be noted that data sets very rarely contain a high enough level of ground truth to effectively use ELM.

The difference in performance is most noticeable for the lower gray scale panel (darker) shown in Figure 7.1. For the brighter panel, Figure 7.2, all of the algorithms share a similar shape with the measured spectrum but differing magnitudes. It should be noted that the shown measured spectrum is an average of all the measured values for the entire calibration panel. The atmospherically compensated spectra reported are for a single pixel on the panel.

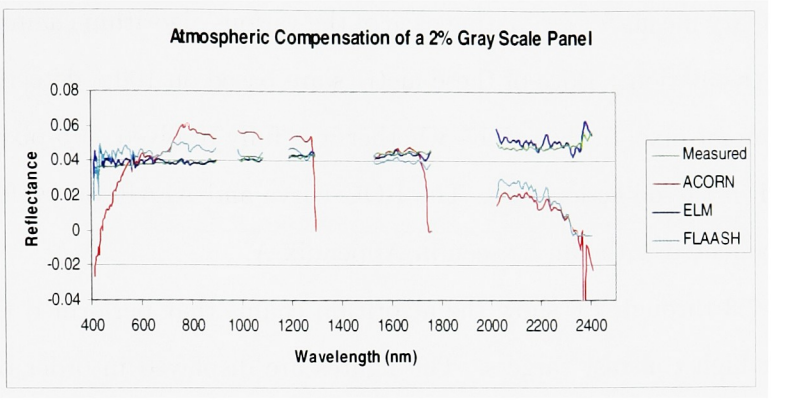

Figure 7.1: Reflectance values, measured and derived by atmospheric compensation for a 2% gray scale panel in the Forest Radiance I scene.

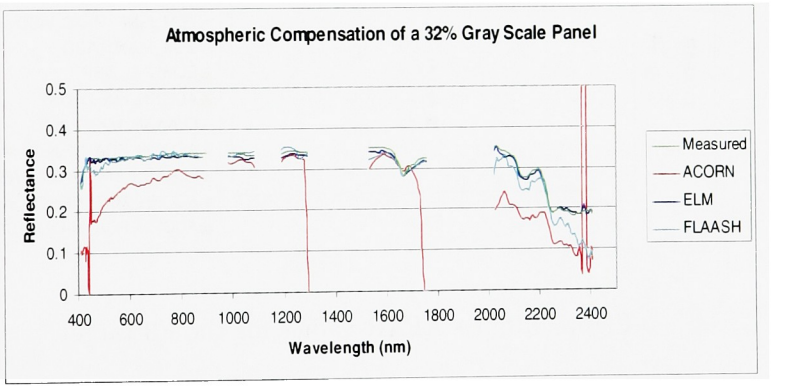

Figure 7.2: Reflectance values, measured and derived by atmospheric compensation for a 32% gray scale panel in the Forest Radiance I scene.

7.2 Algorithm Chain Evaluation

The primary means for evaluating each of the various algorithm chains is via ROC curves and AFAR calculations. Both of these metrics are based on 100% detection. That is to say that every target pixel in the scene - whether it is fully resolved, sub-pixel, glare or shadow - is counted in the detection rate. The ROC curve and AFAR values for every algorithm chain experimented with can be seen in Appendix A.

Figures 7.3 through 7.6 show the algorithm chains that performed the best and worst for the four high contrast targets. The figures are displayed in order of target detection difficulty as determined by the median values (shown in yellow in each figure). The median values are used to determine difficulty because they are directly tied to how easy or difficult the target is to detect. For example, if the median (1 AFAR) value is high, then more algorithm chains had higher overall (1-AFAR) values. If the median value is low, that means that the algorithm chains had a more difficult time detecting the target pixels.

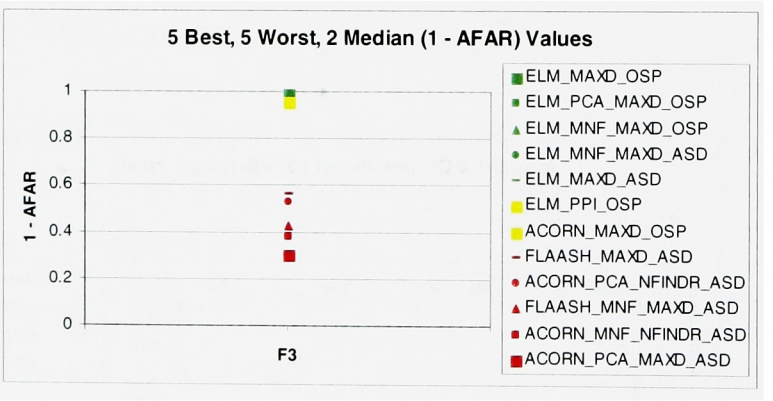

Figure 7.3: 1 AFAR values for the F3 target.

Figures 7.7 through 7.10 show the algorithm chains that performed the best and worst for the four low contrast targets.

The top five and bottom five performers for each target are listed. One observation that immediately presents itself in the above figures is the very wide performance spread that exists between the various algorithm chains for each target. This shows that the choice of

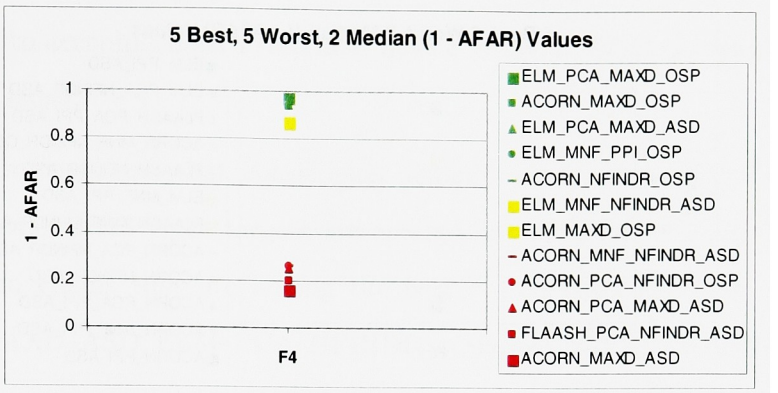

Figure 7.4: 1 - AFAR values for the F4 target.

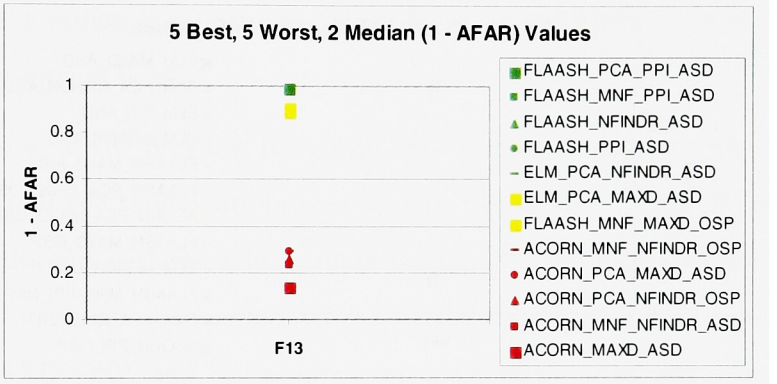

Figure 7.5: 1 - AFAR values for the F13 target.

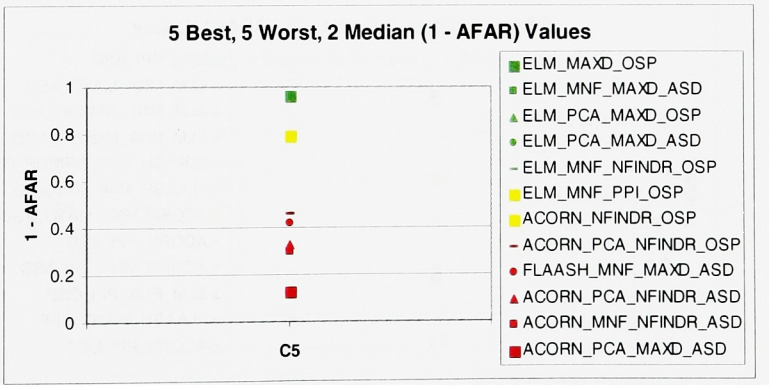

Figure 7.6: 1 - AFAR values for the C5 target.

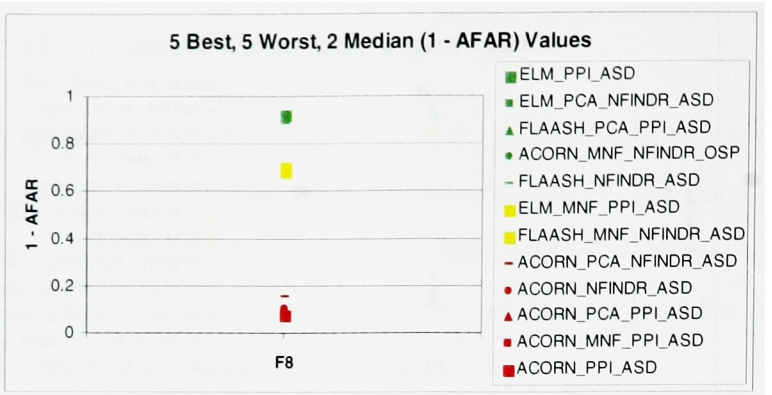

Figure 7.7: 1 - AFAR values for the F8 target.

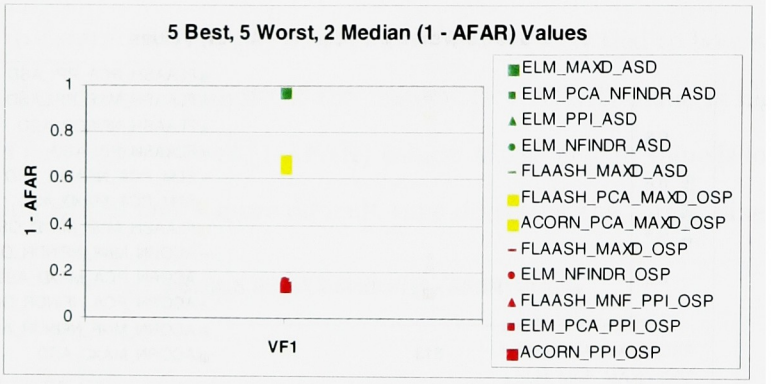

Figure 7.8: 1 - AFAR values for the VF1 target.

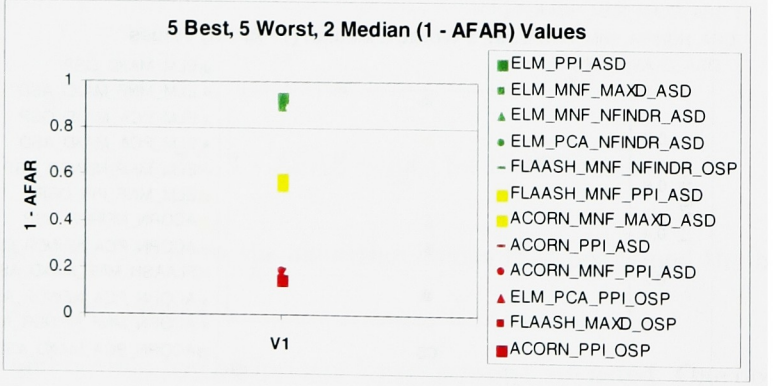

Figure 7.9: 1 - AFAR values for the V1 target.

algorithm used to accomplish each step does have a profound effect on the ability to detect a target in the scene.

There are some chains that consistently appear among either the top or the bottom 5 performers. If all eight targets are treated as one large set, there are three different algorithm chains that are among the best 5 performers for 50% of the targets: ELM–MNF–MAXD–ASD, ELM–PCA–NFINDR–ASD, and ELM–PPI–ASD. Of these three chains, the only common links are the atmospheric compensation method (ELM) and the matched filter used (ASD). Similarly, there are two algorithm chains that performed among the worst 5 for 50% of the targets: ACORN–MNF–NFINDR–ASD and ACORN–PCA–MAXD -ASD. Again, the atmospheric compensation algorithm (ACORN) and the matched filter (ASD) are the common links.

The discussion in the previous paragraph was focused on treating all eight targets as one large group. If the targets are broken up into two smaller groups (high and low contrast), the results are perhaps a little more telling. Of the high contrast targets, there were no algorithm chains that performed among the top 5 for all four targets. One chain, ELM–PCA–MAXD–OSP, was in the top 5 for 3 out of the 4 targets. There were, however, 2 algorithm chains in the bottom 5 performers for all four of the high contrast targets: ACORN–MNF–NFINDR–ASD and ACORN–PCA–MAXD–ASD. The ACORN–PCA–NFINDR

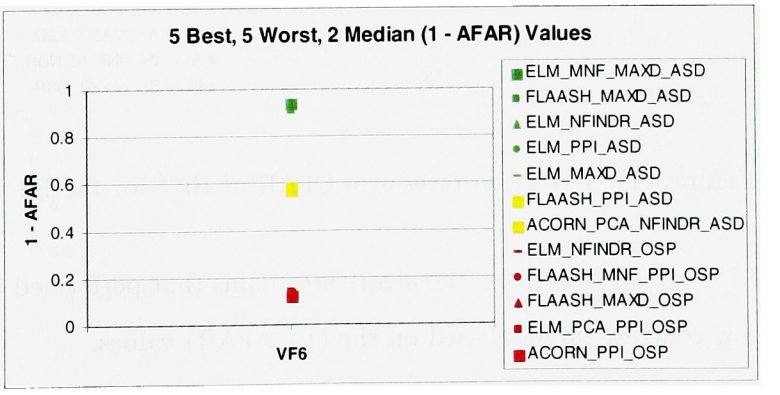

Figure 7.10: 1 - AFAR values for the VF6 target.

-OSP chain is among the bottom 5 performers for 3 of the 4 high contrast targets.

The low contrast targets exhibit similar phenomenons to the high contrast targets. One algorithm chain, ELM–PPI–ASD, was in the top 5 performers for all of the low contrast targets, and the ELM–PCA–NFINDR–ASD chain was in the top 5 for 3 of the 4 targets. There were no algorithm chains among the bottom 5 for all of the low contrast targets, but 3 chains were in the bottom 5 for 3 of the targets: ACORN–PPI–ASD, ELM–PCA–PPI–OSP, and FLAASH–MAXD–OSP.

Another way to evaluate the performance of the various algorithm chains is to look at the detection rates at a constant false alarm rate (CFAR). This statistic shows how "quickly" an algorithm chain is able to detect the target pixels. A high detection rate means that more target pixels were found prior to a set number of false alarms than an algorithm with a low detection rate. Figures 7.11 through 7.18 below show the 5 highest, 5 lowest, and 2 median detection rates for all of the targets at a constant false alarm rate of 10^{-4}. The targets are displayed in the same order as the (1- AFAR) figures presented earlier.

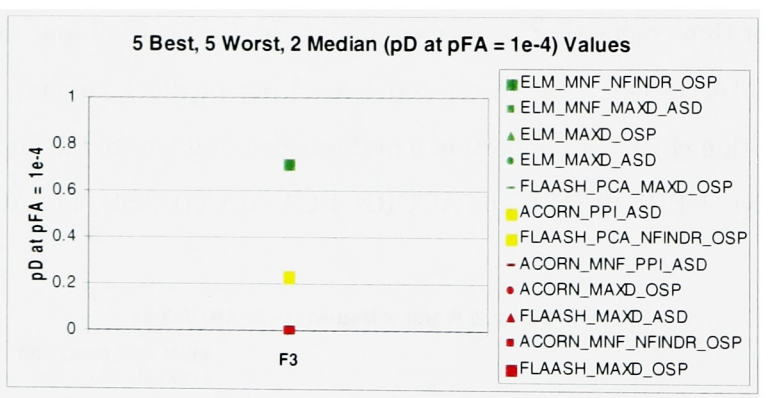

Figure 7.11: Detection rates at a CFAR of 10^{-4} for the F3 target.

Figures 7.15 through 7.18 show the algorithm chains that performed the best and worst for the four low contrast targets based on the (1 - AFAR) values.

It is immediately obvious that the targets that had high (1 AFAR) values do not necessarily have the best CFAR detection rates. This shows that the "best" algorithm

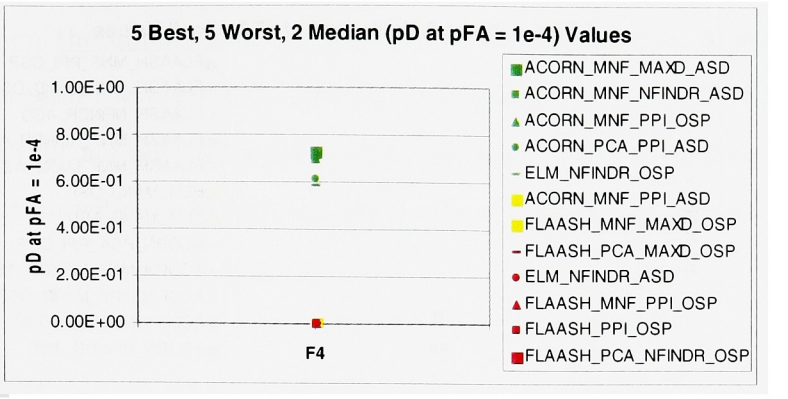

Figure 7.12: Detection rates at a CFAR of 10^{-4} for the F4 target.

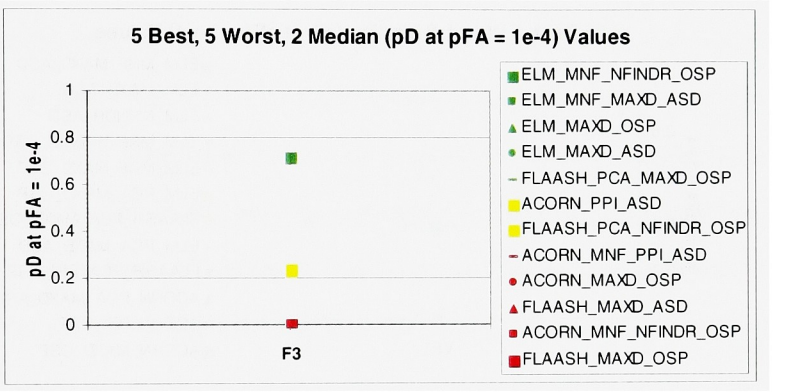

Figure 7.13: Detection rates at a CFAR of 10^{-4} for the F13 target.

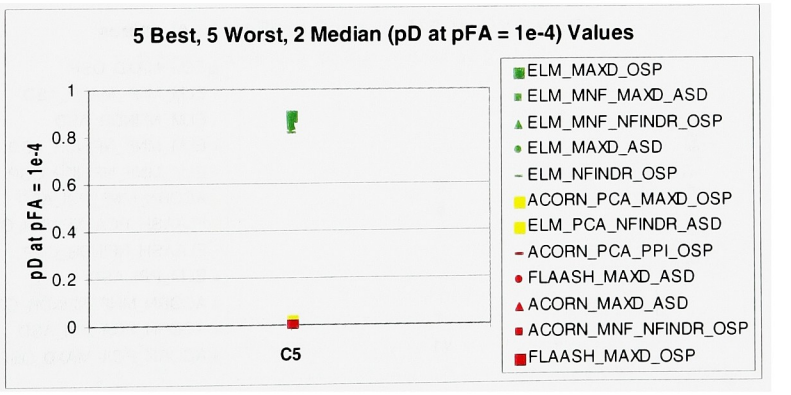

Figure 7.14: Detection rates at a CFAR of 10^{-4} for the C5 target.

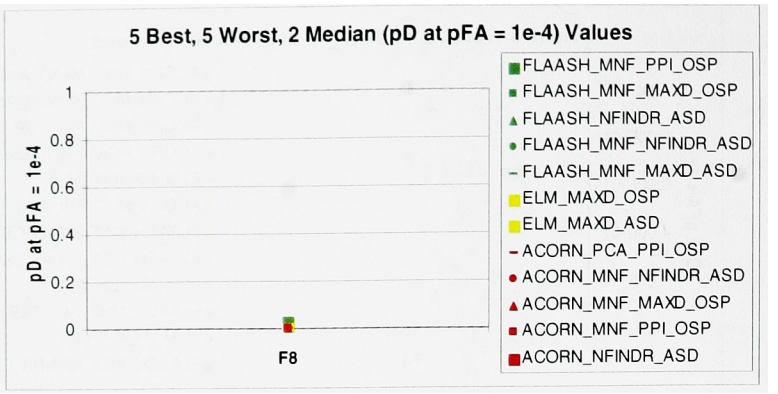

Figure 7.15: Detection rates at a CFAR of 10^{-4} for the F8 target.

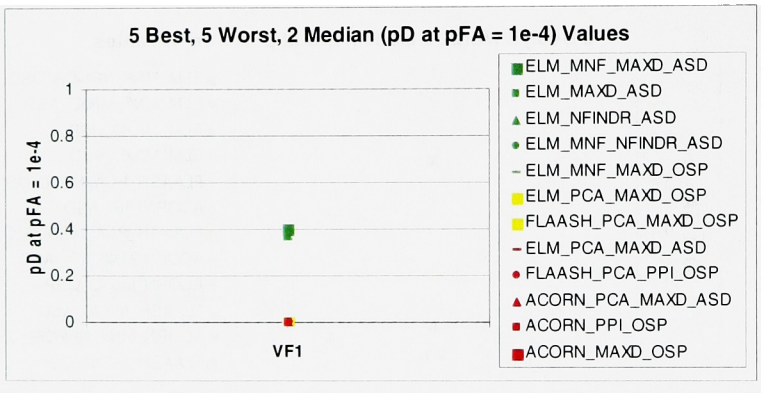

Figure 7.16: Detection rates at a CFAR of 10^{-4} for the VF1 target.

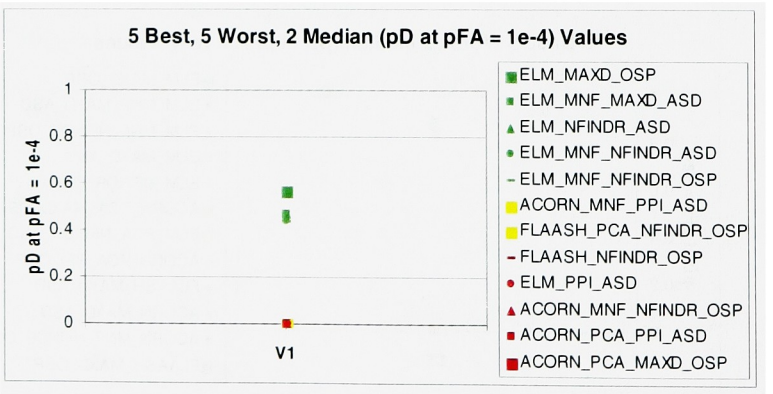

Figure 7.17: Detection rates at a CFAR of 10^{-4} for the V1 target.

chain depends on the application. If detection of *all* target pixels is desired, then the algorithm chain with the highest (1 - AFAR) rate would be the "best". If detection of a high number of (but not all) target pixels and a low number of false alarms is desired, then the algorithm chain with the highest detection rate at a low false alarm rate would be the "best".

7.2.1 Average (1 - AFAR) Values

The previous discussion focuses on the (1 - AFAR) value and the CFAR detection rate for each individual algorithm chain in order to help describe algorithm chain performance. It is also important to look at the overall performance of each algorithm to help establish a good baseline to use when first attempting target detection on a new data set. In order to accomplish this, the algorithm chains were grouped by the atmospheric compensation algorithm that began each chain, and the average (1 - AFAR) value was calculated for each group over all of the targets. The result was that the chains incorporating ELM drastically outperformed the chains using FLAASH or ACORN. The results of this grouping can be seen in Table 7.1.

Once the top performing atmospheric compensation algorithm was determined, then each algorithm chain using the top atmospheric compensation algorithm was then grouped

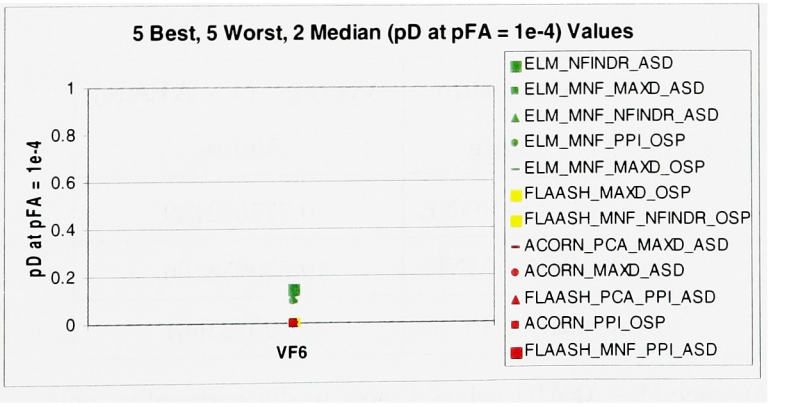

Figure 7.18: Detection rates at a CFAR of 10^{-4} for the VF6 target.

by its dimensionality/noise reduction algorithm. The average (1 AFAR) value was again calculated. As can be seen in Table 7.2, PCA only slightly outperformed MNF, but both were considerably better than doing no dimensionality/noise reduction at all.

Next, the top performing chains, the ones containing ELM and PCA as the first two steps, were grouped by the next link in the algorithm chain: endmember selection. Table 7.3 shows the results of this next level of grouping.

Again, there were two algorithms that resulted in very similar average (1 - AFAR) values, MAXD and N-FINDR. Both of these algorithms had significantly higher values than PPI. Since N-FINDR was the top performer, the next step was to look at the average (1 AFAR) values of the chains using ELM, PCA, and N-FINDR grouped by the matched filter used. Table 7.4 shows the results of this grouping.

As Table 7.4 shows, the chain using ASD greatly outperformed the chain using OSP.

Atmospheric Compensation	**Average (1 - AFAR) Value**
ACORN	0.505990635
ELM	0.805887969
FLAASH	0.71301075

Table 7.1: Average (1 AFAR) values based on atmospheric compensation algorithm.

Algorithm Chain	**Average (1 - AFAR) Value**
ELM / NONE	0.777652139
ELM / MNF	0.819258106
ELM / PCA	0.820753661

Table 7.2: Average (1 AFAR) values based on dimensionality/noise reduction algorithm for chains using ELM.

The overriding conclusion that can be drawn from this analysis is that the algorithm chain ELM-PCA-NFINDR-ASD is the best performing chain *on average.*

As mentioned in Section 7.1, ELM is not very likely to be used in a practical application because of its reliance on accurate ground truth data. For the purposes of this research, ELM was treated as being equally likely to be used as both FLAASH and ACORN. So, the question of which algorithm chain would perform the best if ELM was not a viable option for atmospheric compensation can be raised. According to the results shown in Table 7.1, one would expect that FLAASH would always be a better option than ACORN because of the disparity in their average (1 AFAR) values. By and large, this statement holds true. Naturally, there are a few data points contradictory to this assertion. For example, the second best performing algorithm chain for the F4 target (Figure 7.4) based on (1 AFAR) value uses ACORN as its atmospheric compensation routine. However, it can be concluded that chains using FLAASH outperform chains using ACORN the vast majority of the time.

Algorithm Chain	**Average (1 - AFAR) Value**
ELM / PCA / MAXD	0.87428808
ELM / PCA / N-FINDR	0.876219555
ELM / PCA / PPI	0.711753348

Table 7.3: Average (1 AFAR) values based on endmember selection algorithm for chains using ELM and PCA.

Algorithm Chain	**Average (1 - AFAR) Value**
ELM / PCA / N-FINDR / ASD	0.923335609
ELM / PCA / N-FINDR / OSP	0.829103502

Table 7.4: Average (1 AFAR) values based on matched filter algorithm for chains using ELM, PCA, and N-FINDR.

7.3 Conclusions and Future Work

It is difficult to make any broad conclusions based on the results of this research. First of all, only a very small piece of what could be a much larger project was looked at. There are a significant number of algorithms for each step in the target detection chain that were not addressed and new algorithms are being developed all the time.

One of the main points that can be made is concerning the effect of the different algorithms for each step. It was expected that different results would be seen for the different chains being used. What was seen was that there is a very large range of performance from very good to very poor for different combinations of algorithms.

It would be very beneficial to continue this research by expanding the number of algorithms tested for each step in the target detection chain. Also, repeating this research on differing scenes, *i.e.* a cluttered urban scene, would be a step toward being able to make much more broad generalizations about the effectiveness of specific algorithm chains in a given situation.

Missing Page

Appendix A

ROC Curves and AFAR Data

The following figures are all of the graphs generated in conjunction with this research. All of the discussion associated with these figures is in Section 7. The graphs are sorted by target and each ROC curve and AFAR data set is labeled using the names of the algorithms in the chain used to generate them. The colors used are associated with the endmember selection algorithms: red MAXD, blue - NFINDR, cyan PPI.

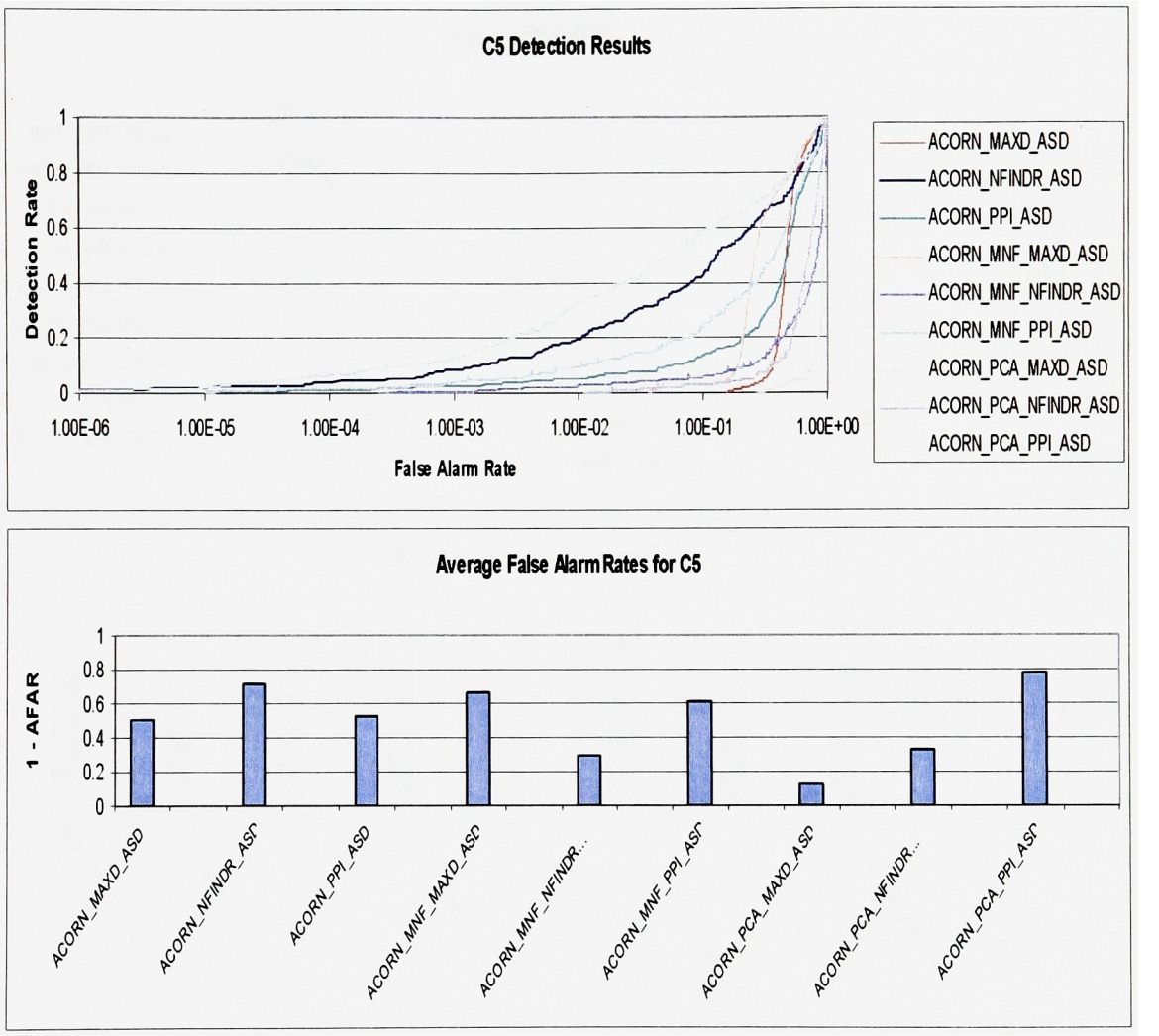

Figure A.1: ROC and (1 − AFAR) graphs for the C5 target that employed the ACORN atmospheric compensation algorithm and the ASD matched filter.

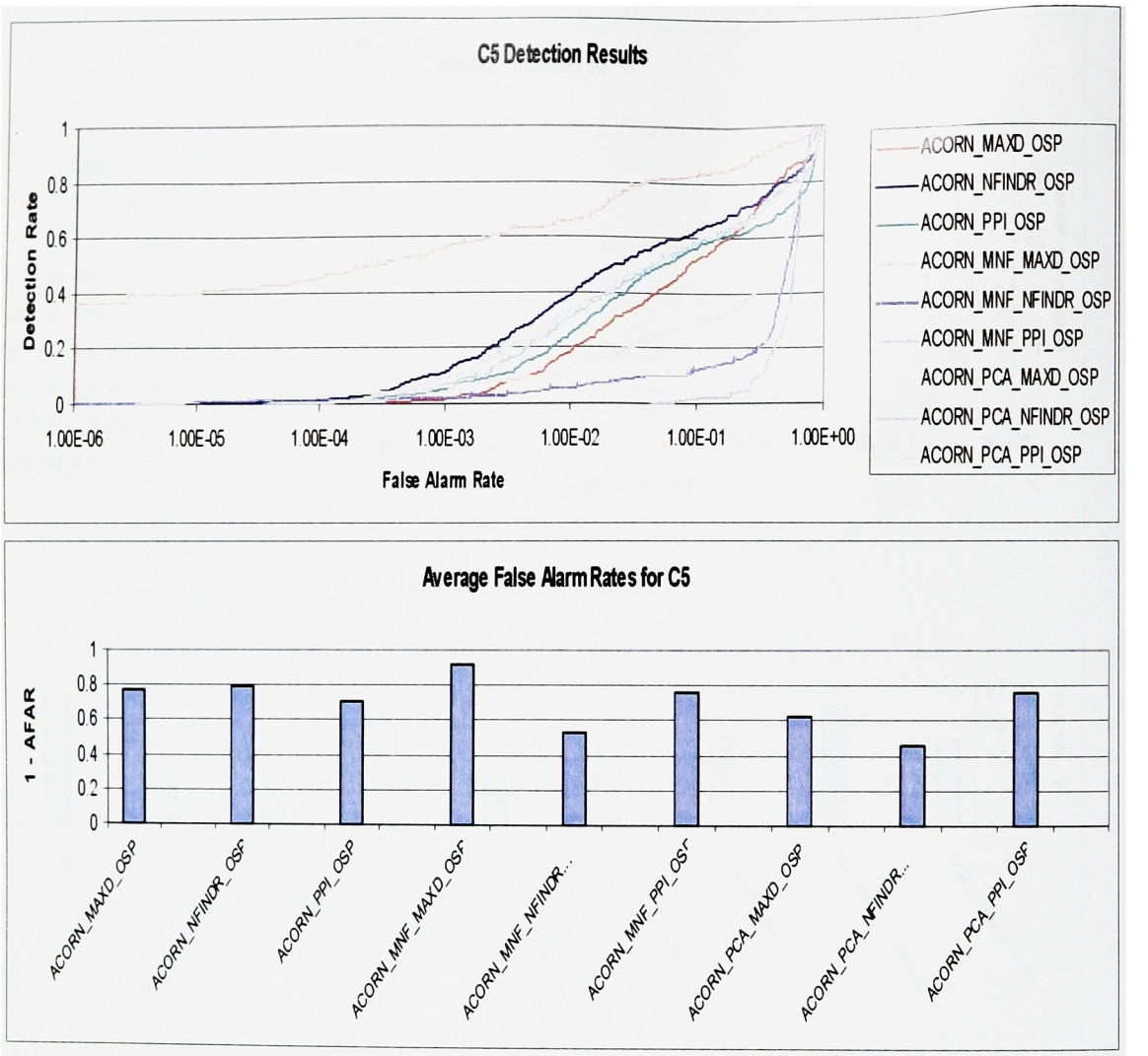

Figure A.2: ROC and (1 – AFAR) graphs for the C5 target that employed the ACORN atmospheric compensation algorithm and the OSP matched filter.

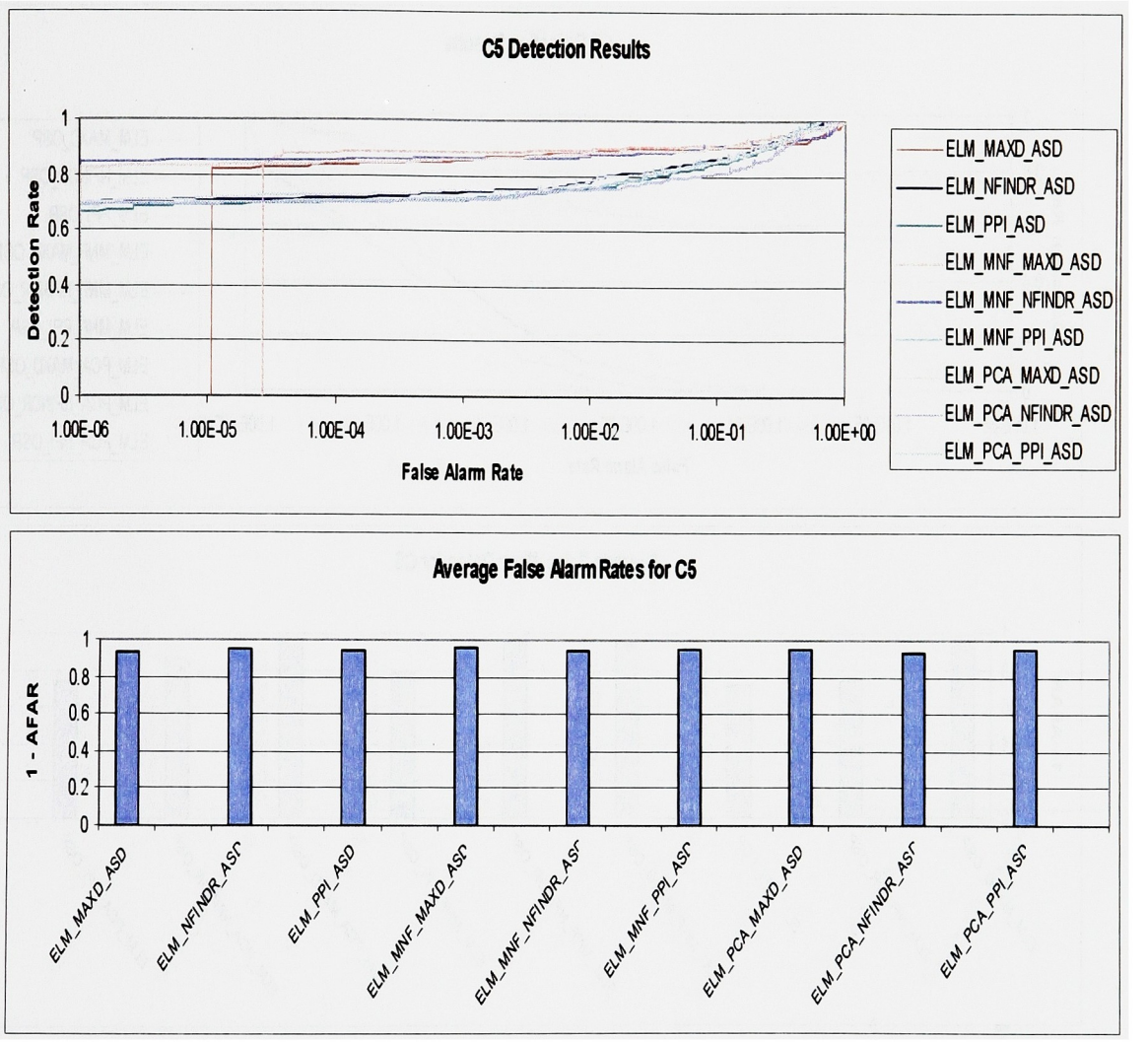

Figure A.3: ROC and (1 – AFAR) graphs for the C5 target that employed the ELM atmospheric compensation algorithm and the ASD matched filter.

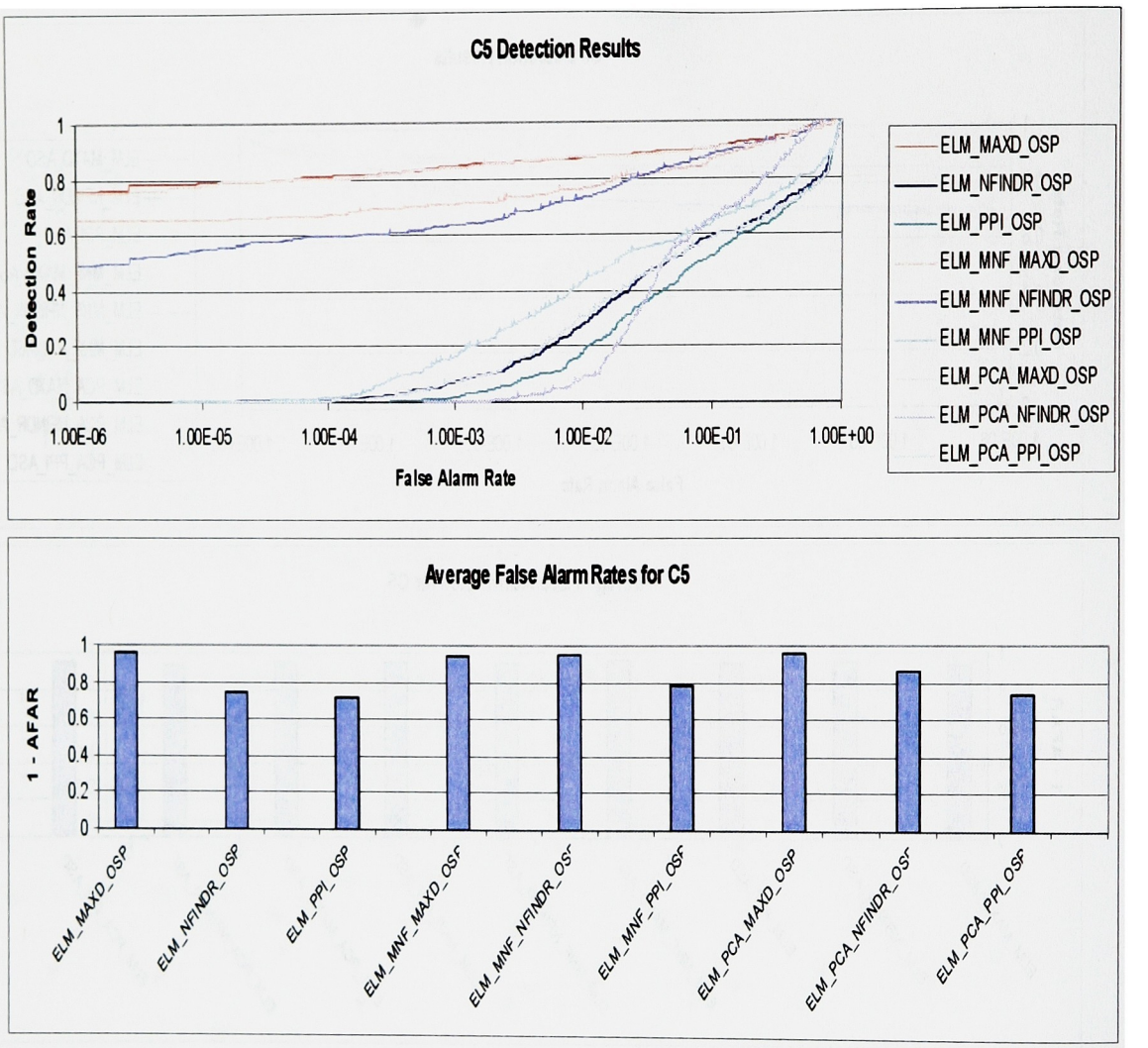

Figure A.4: ROC and (1 − AFAR) graphs for the C5 target that employed the ELM atmospheric compensation algorithm and the OSP matched filter.

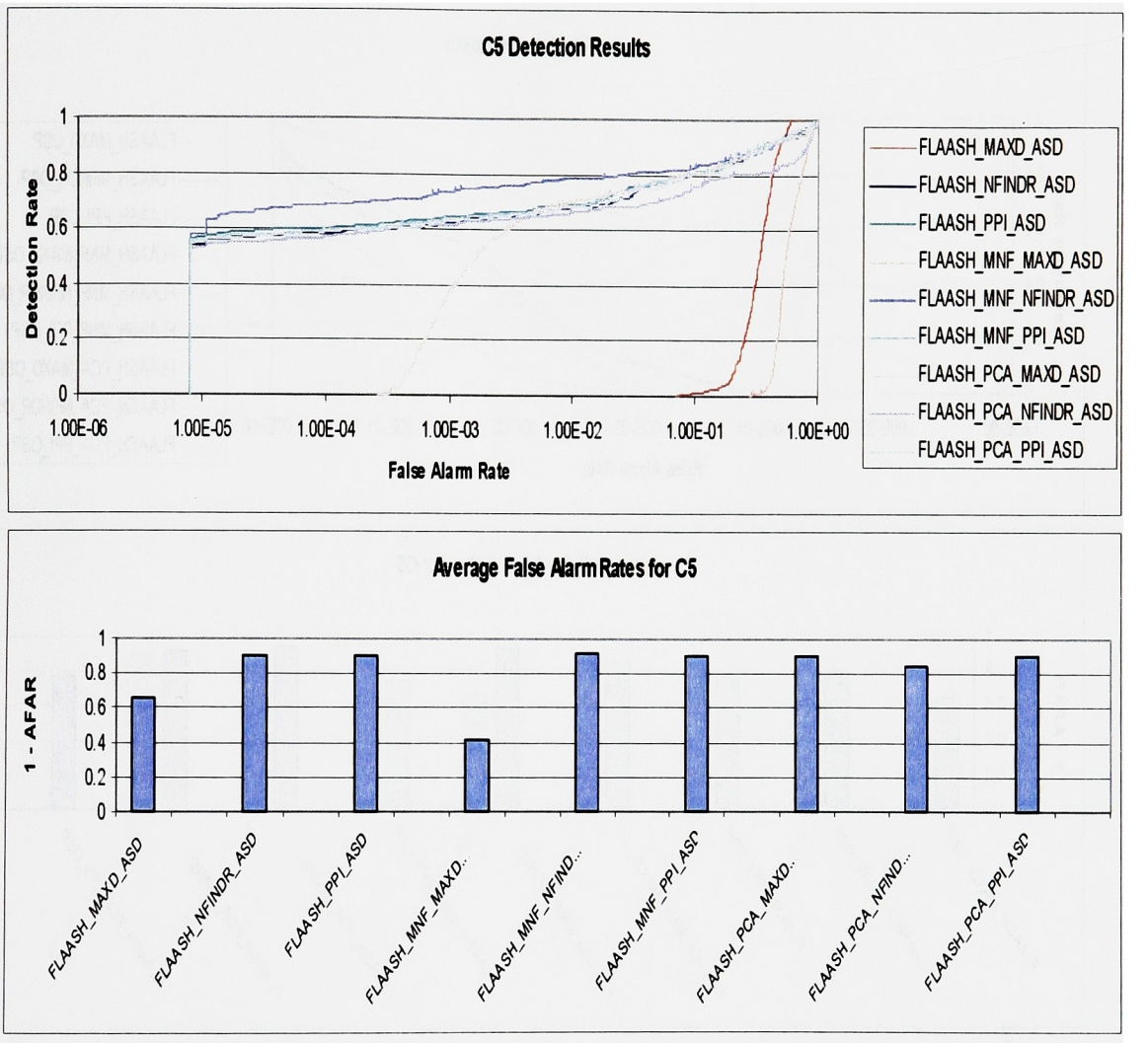

Figure A.5: ROC and (1 – AFAR) graphs for the C5 target that employed the FLAASH atmospheric compensation algorithm and the ASD matched filter.

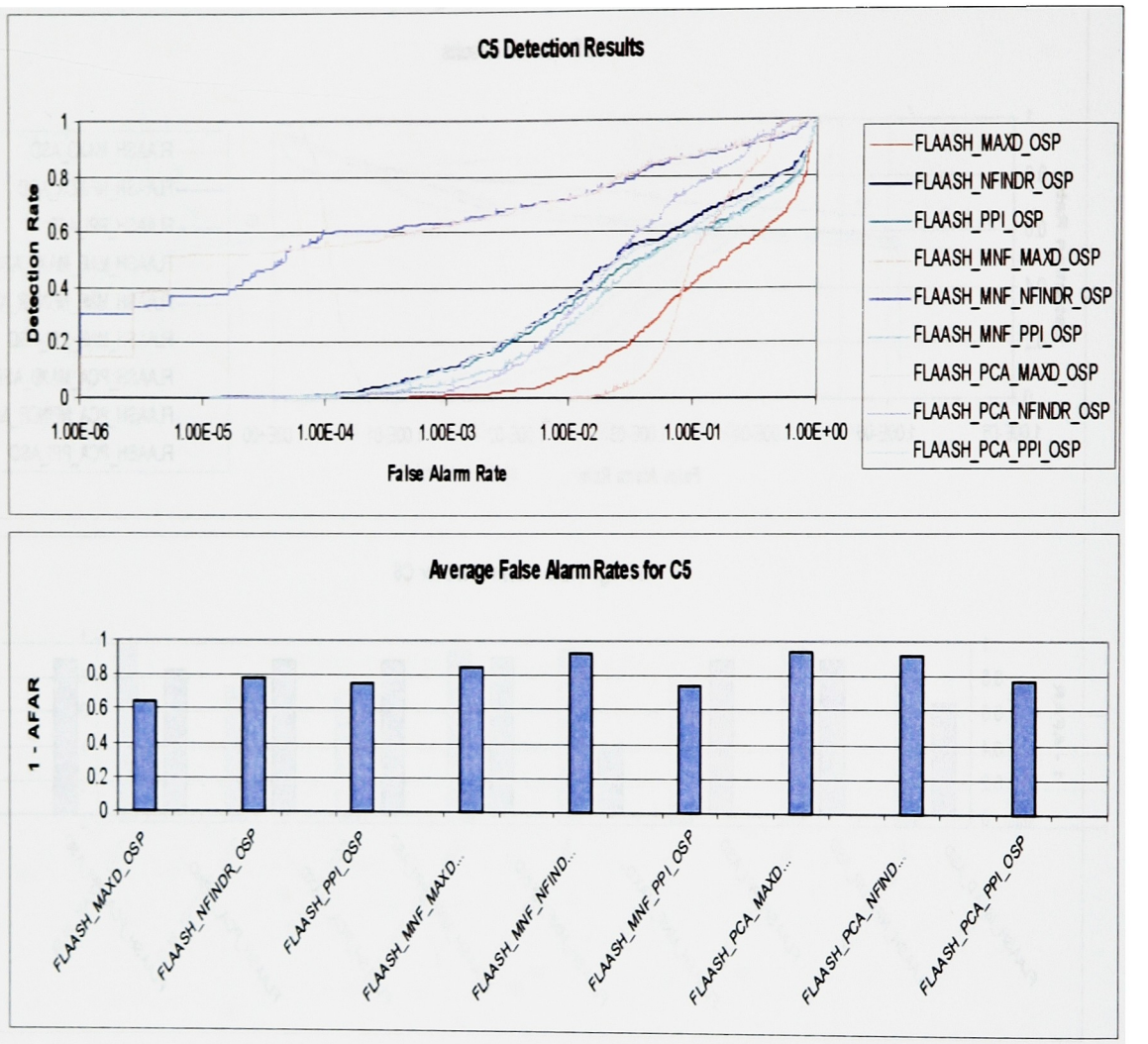

Figure A.6: ROC and (1 – AFAR) graphs for the C5 target that employed the FLAASH atmospheric compensation algorithm and the OSP matched filter.

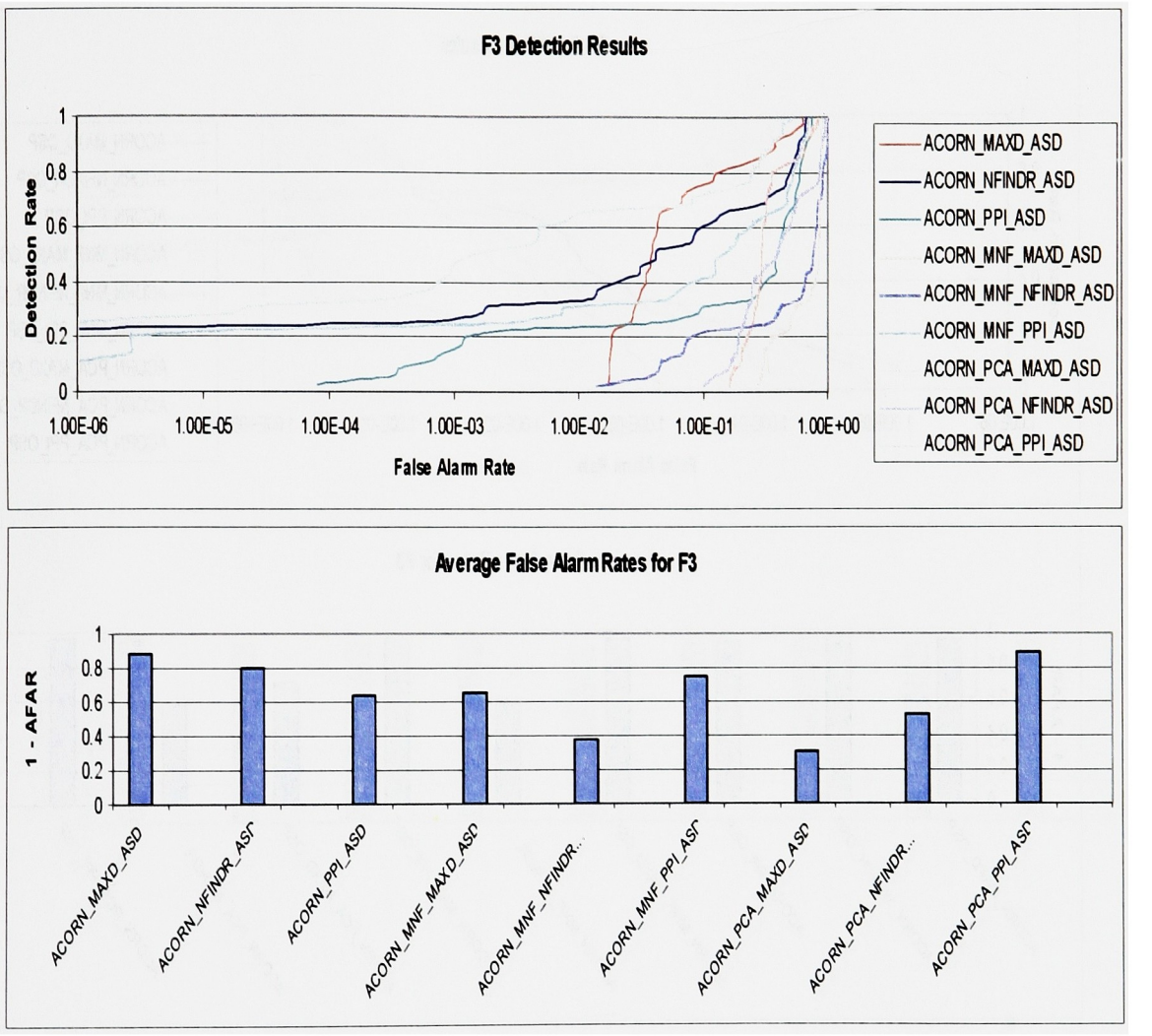

Figure A.7: ROC and (1 – AFAR) graphs for the F3 target that employed the ACORN atmospheric compensation algorithm and the ASD matched filter.

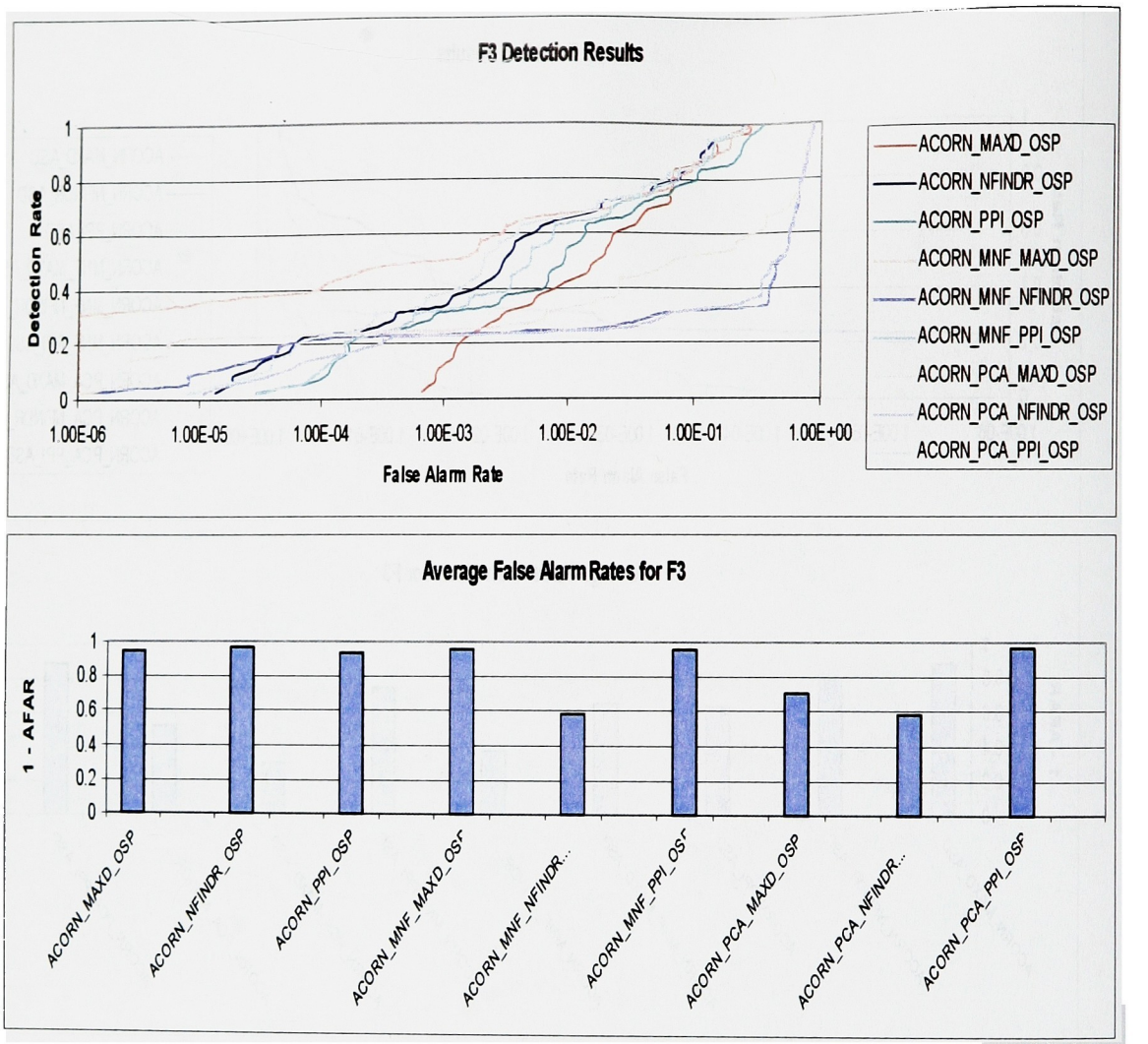

Figure A.8: ROC and (1 – AFAR) graphs for the F3 target that employed the ACORN atmospheric compensation algorithm and the OSP matched filter.

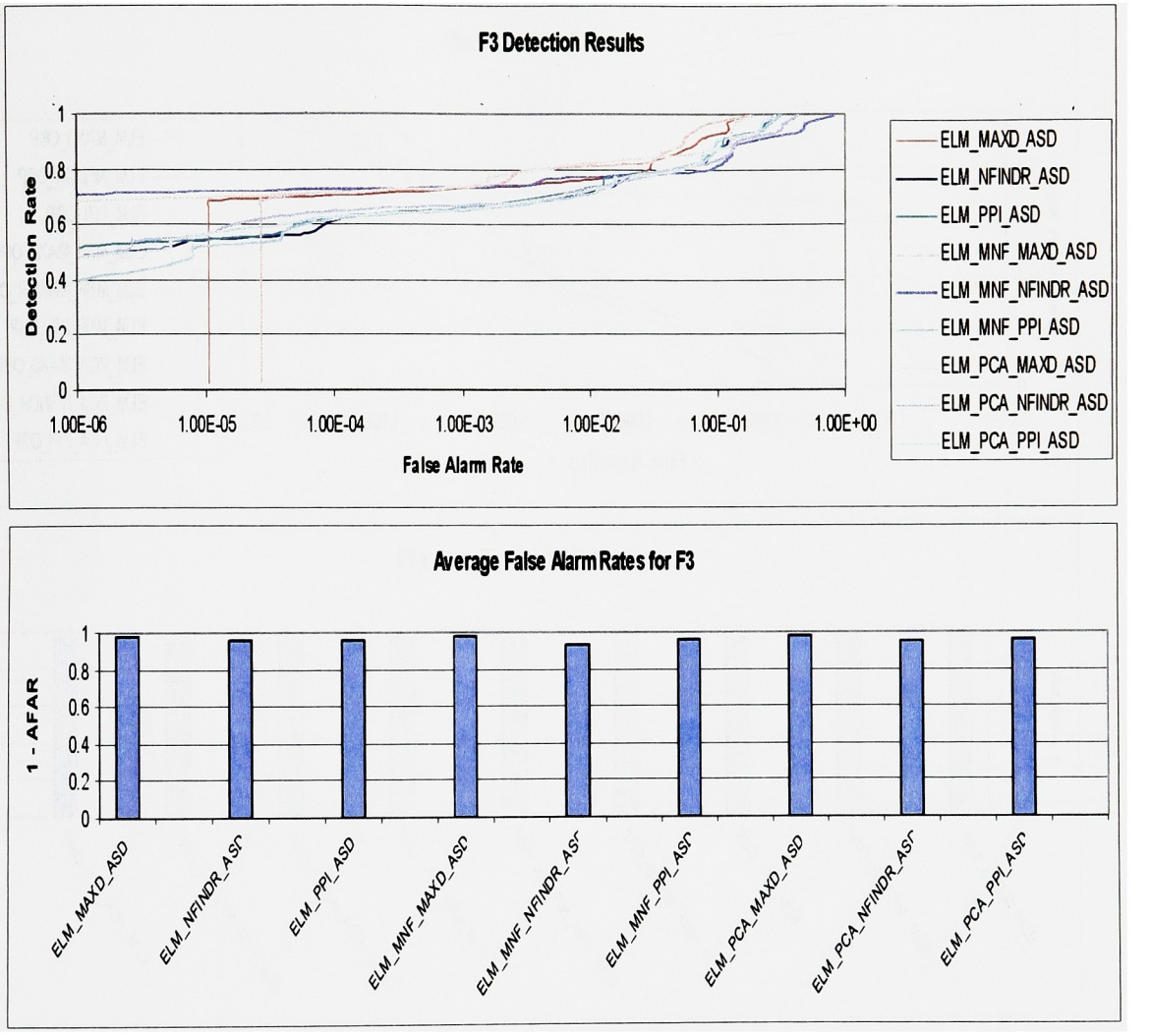

Figure A.9: ROC and (1 – AFAR) graphs for the F3 target that employed the ELM atmospheric compensation algorithm and the ASD matched filter.

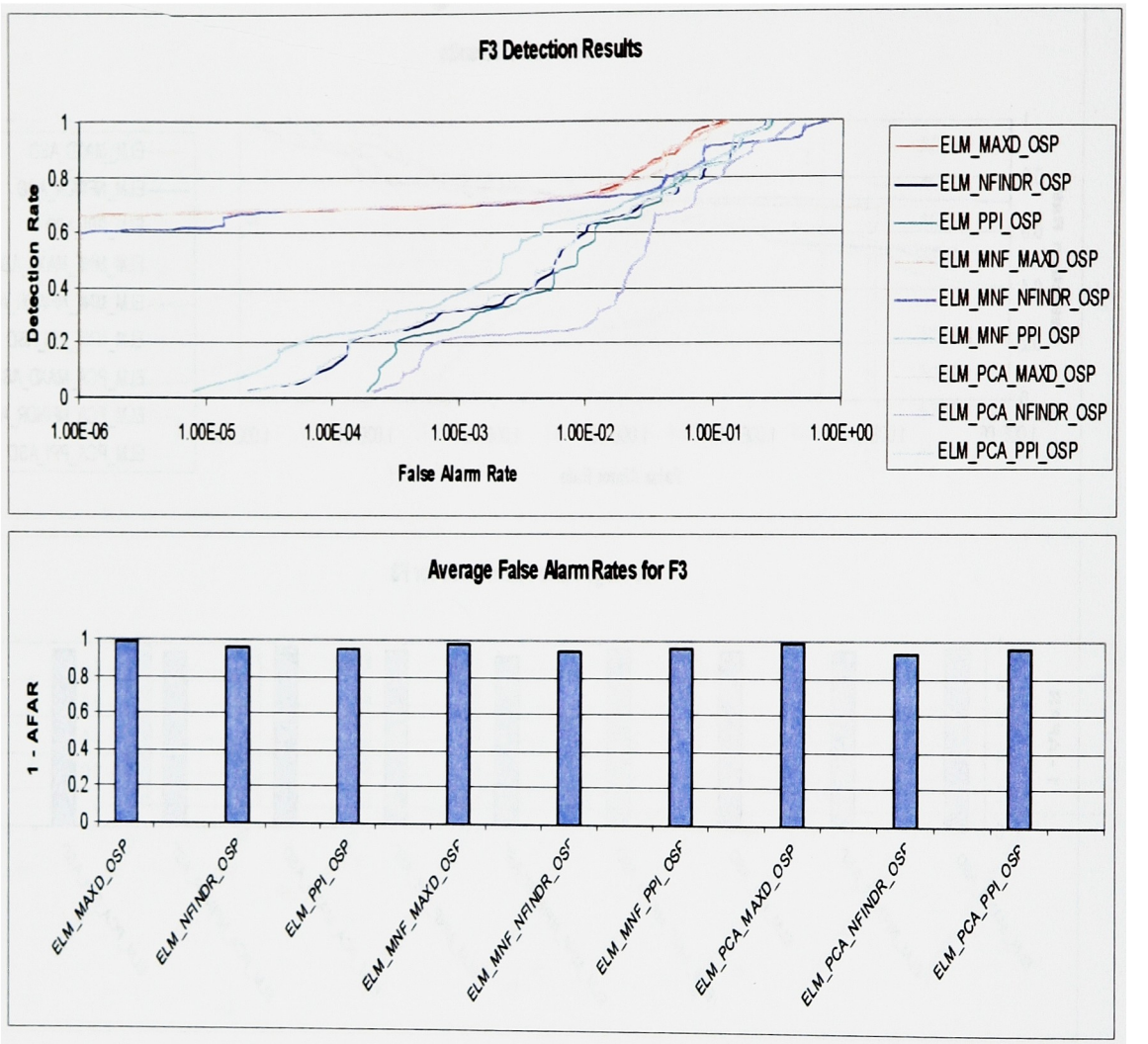

Figure A.10: ROC and (1 − AFAR) graphs for the F3 target that employed the ELM atmospheric compensation algorithm and the OSP matched filter.

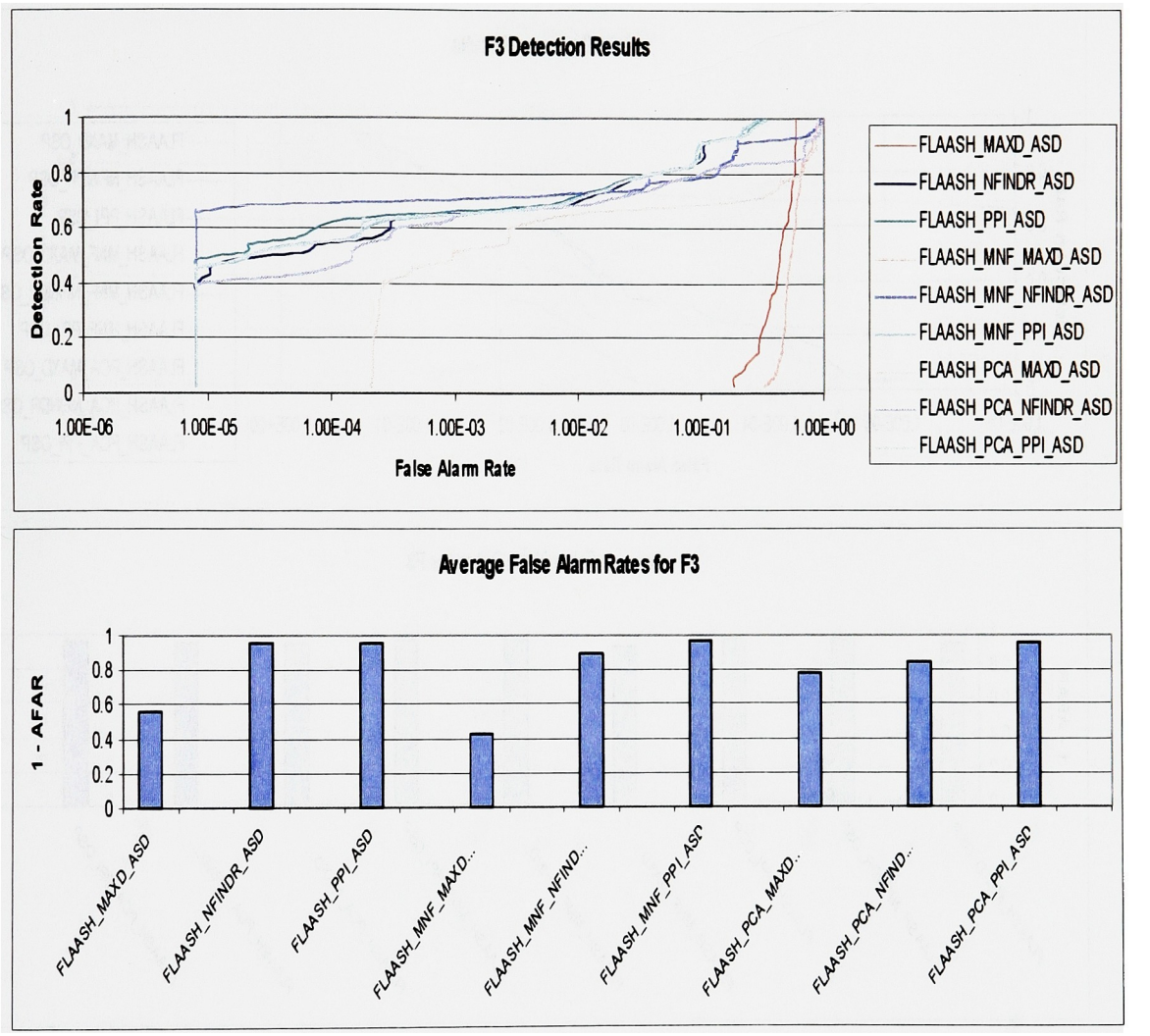

Figure A.11: ROC and (1 – AFAR) graphs for the F3 target that employed the FLAASH atmospheric compensation algorithm and the ASD matched filter.

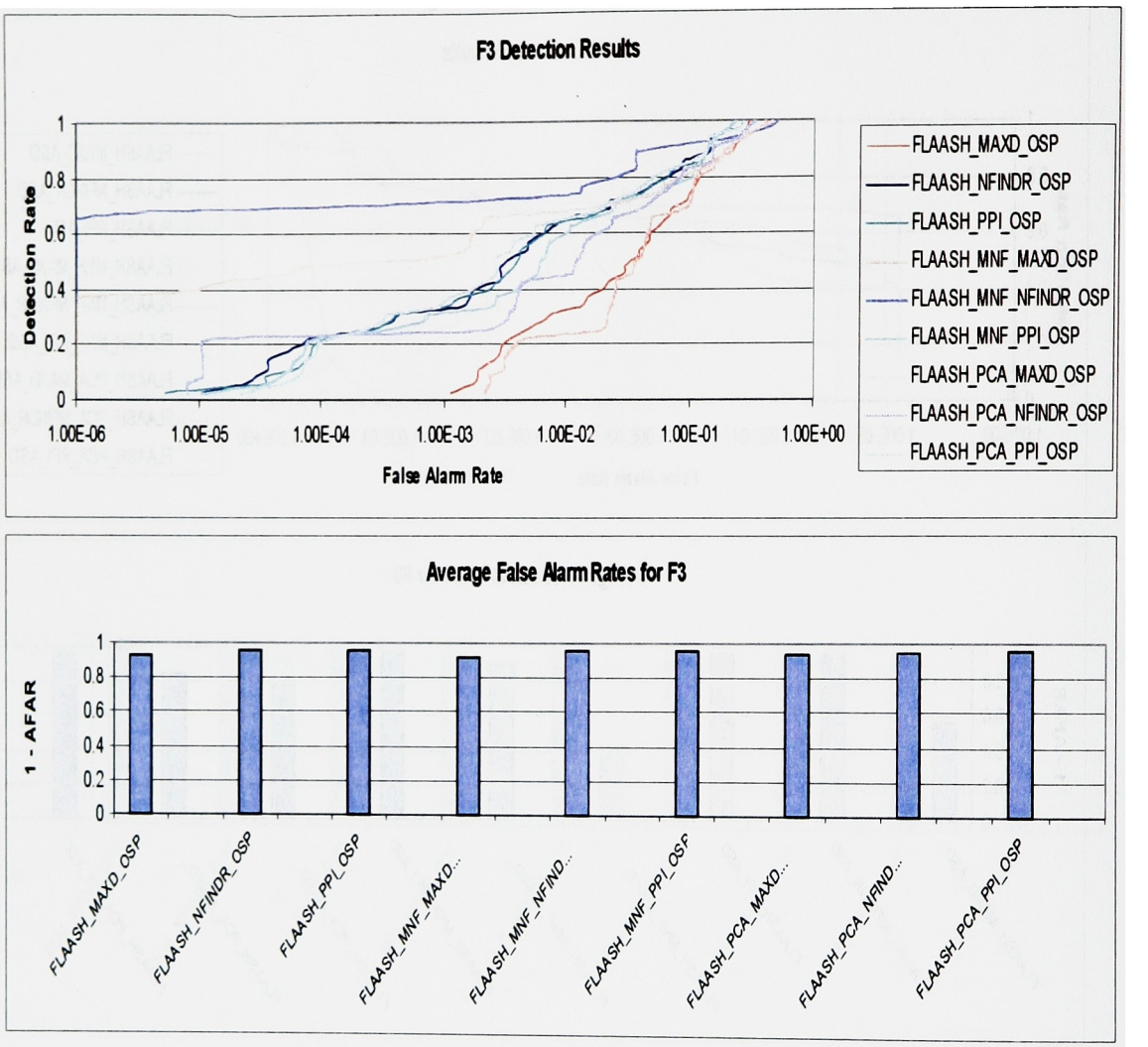

Figure A.12: ROC and (1 – AFAR) graphs for the F3 target that employed the FLAASH atmospheric compensation algorithm and the OSP matched filter.

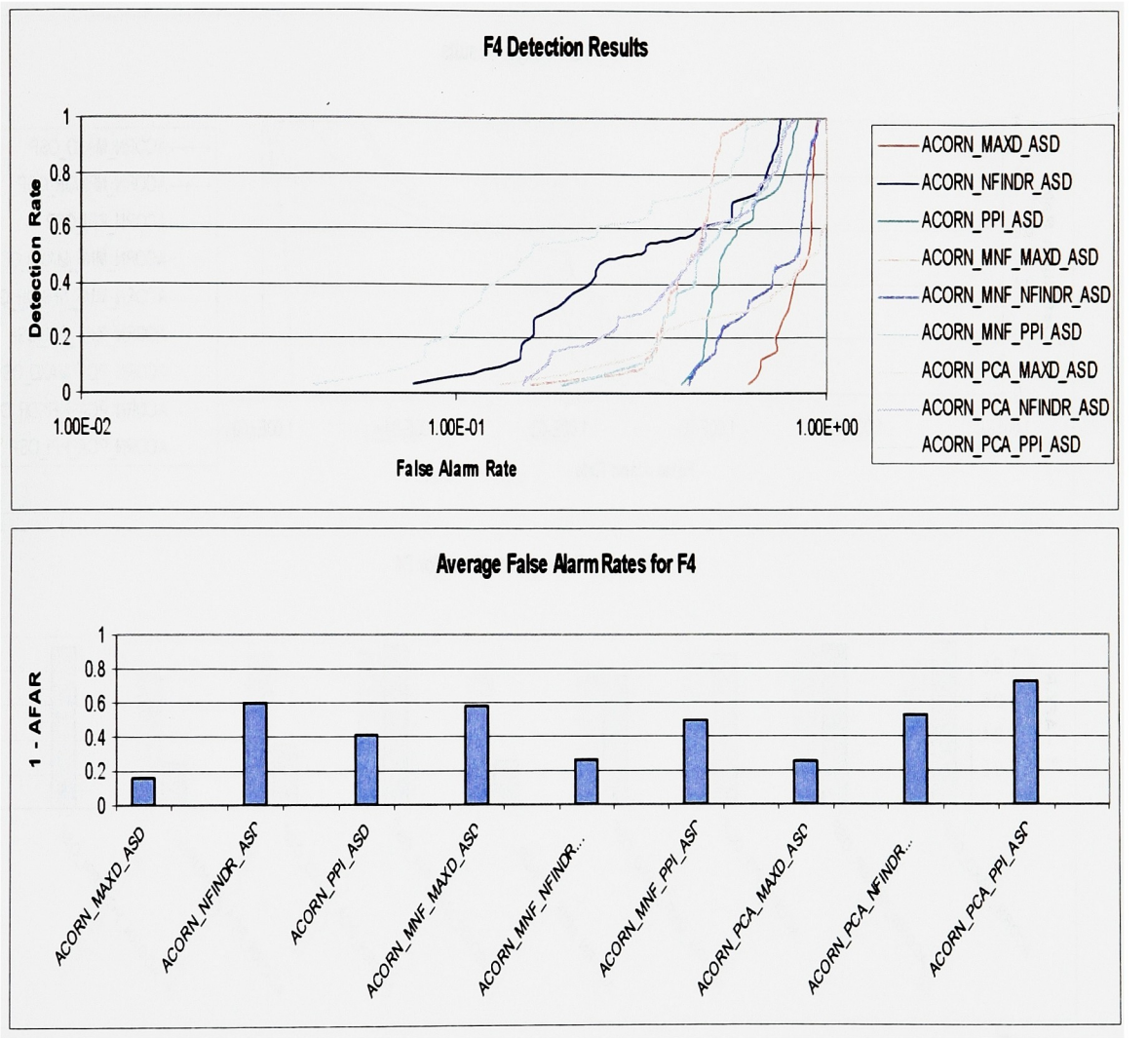

Figure A.13: ROC and (1 − AFAR) graphs for the F4 target that employed the ACORN atmospheric compensation algorithm and the ASD matched filter.

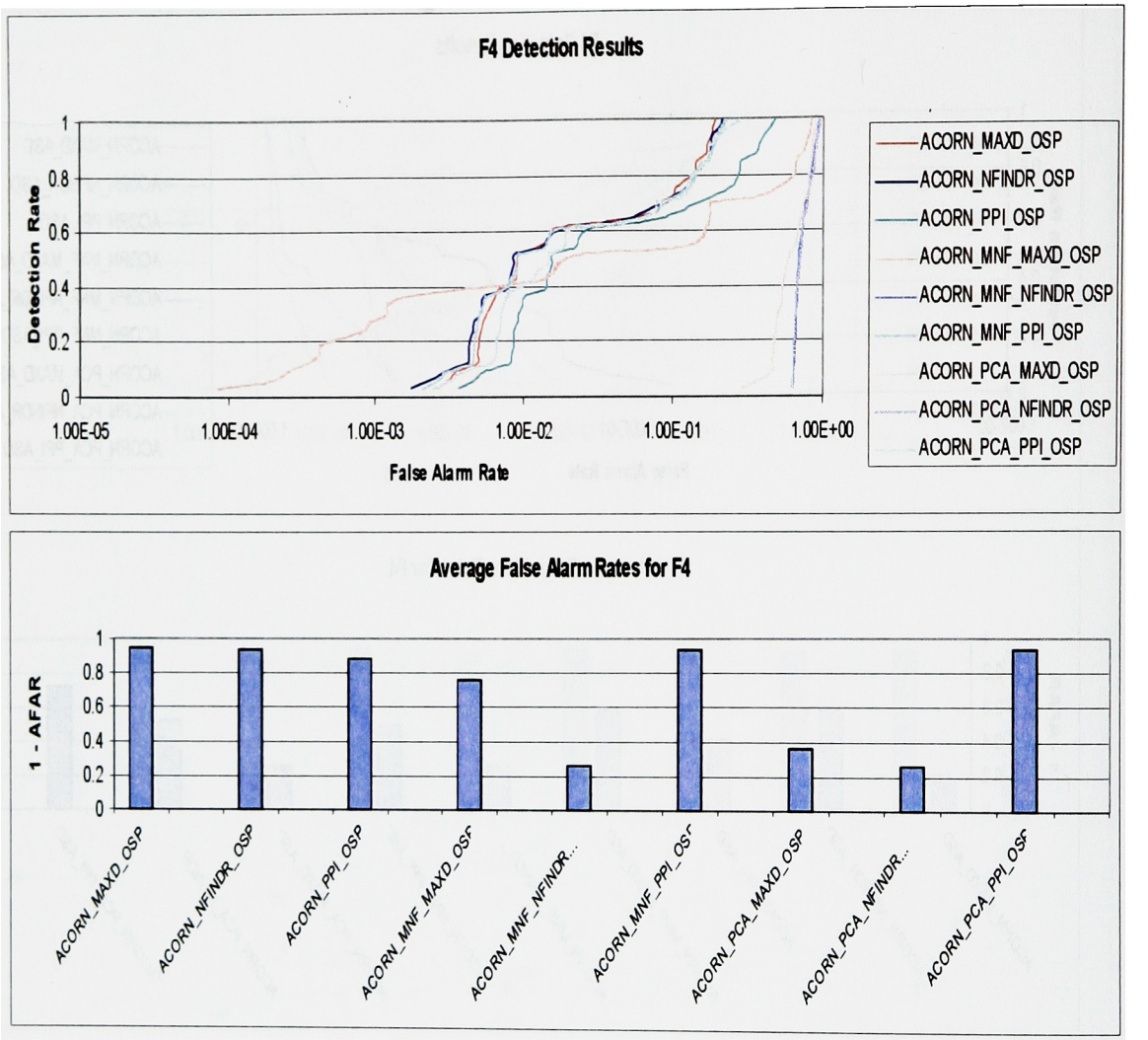

Figure A.14: ROC and (1 – AFAR) graphs for the F4 target that employed the ACORN atmospheric compensation algorithm and the OSP matched filter.

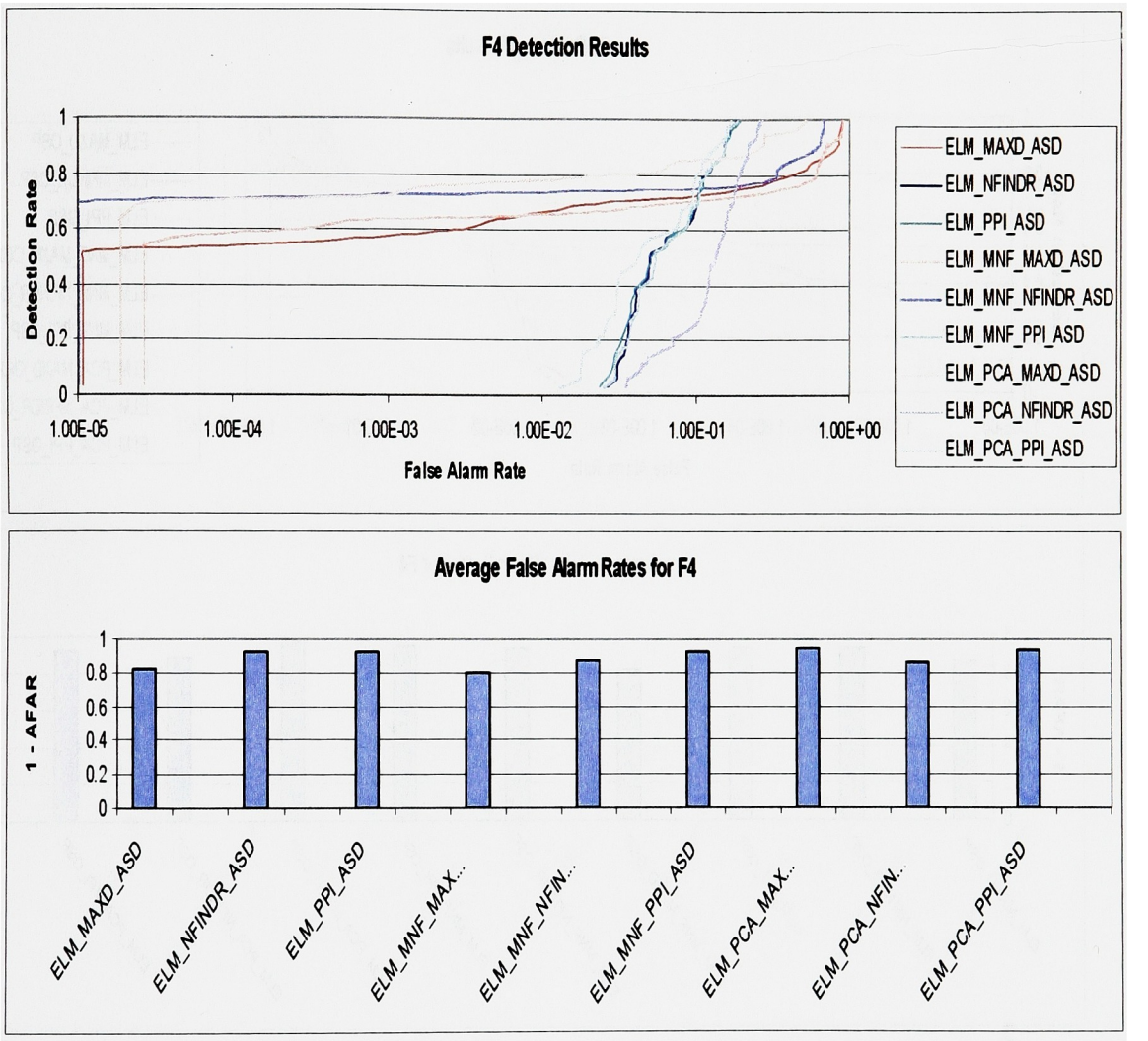

Figure A.15: ROC and (1 – AFAR) graphs for the F4 target that employed the ELM atmospheric compensation algorithm and the ASD matched filter.

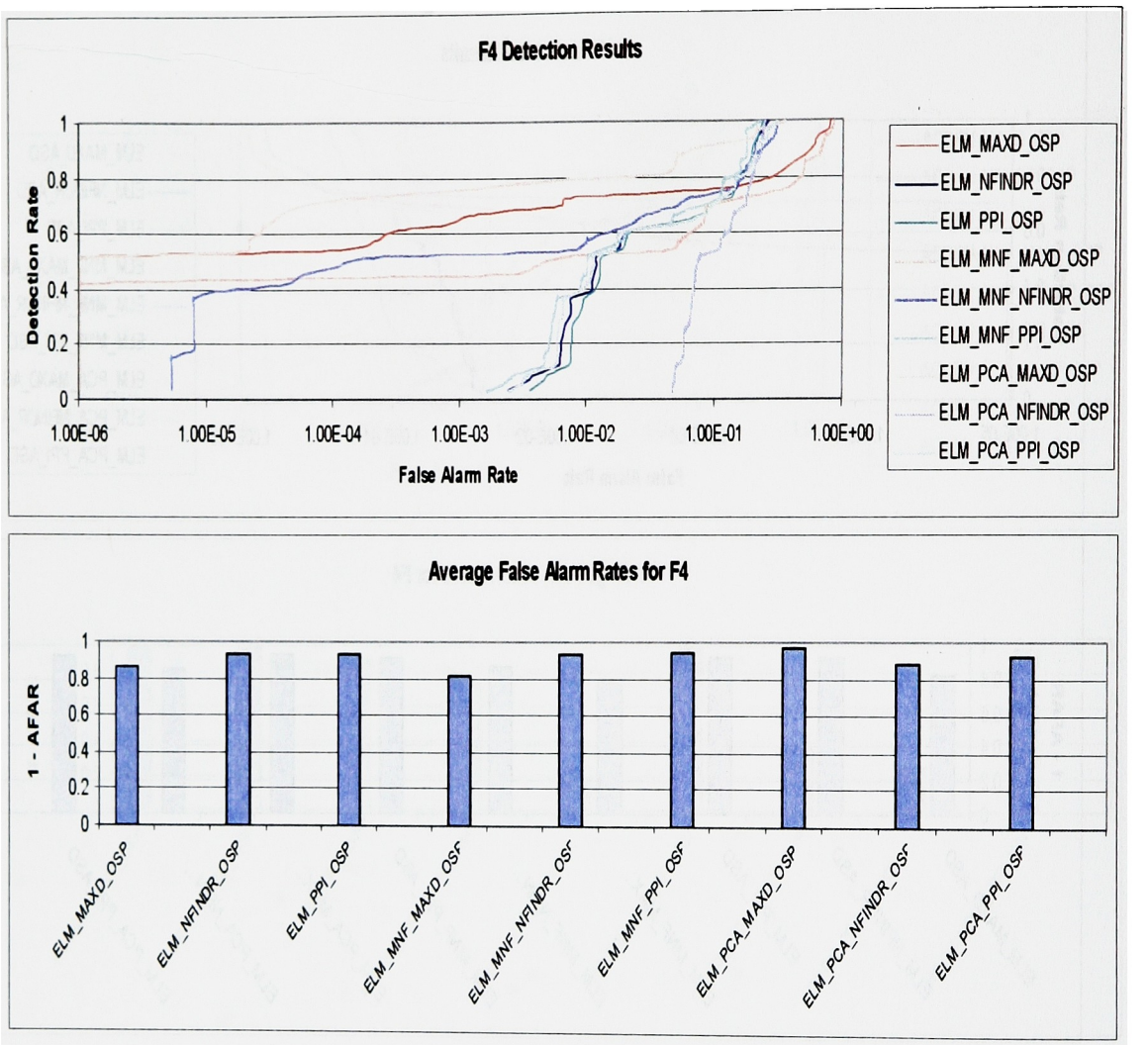

Figure A.16: ROC and (1 − AFAR) graphs for the F4 target that employed the ELM atmospheric compensation algorithm and the OSP matched filter.

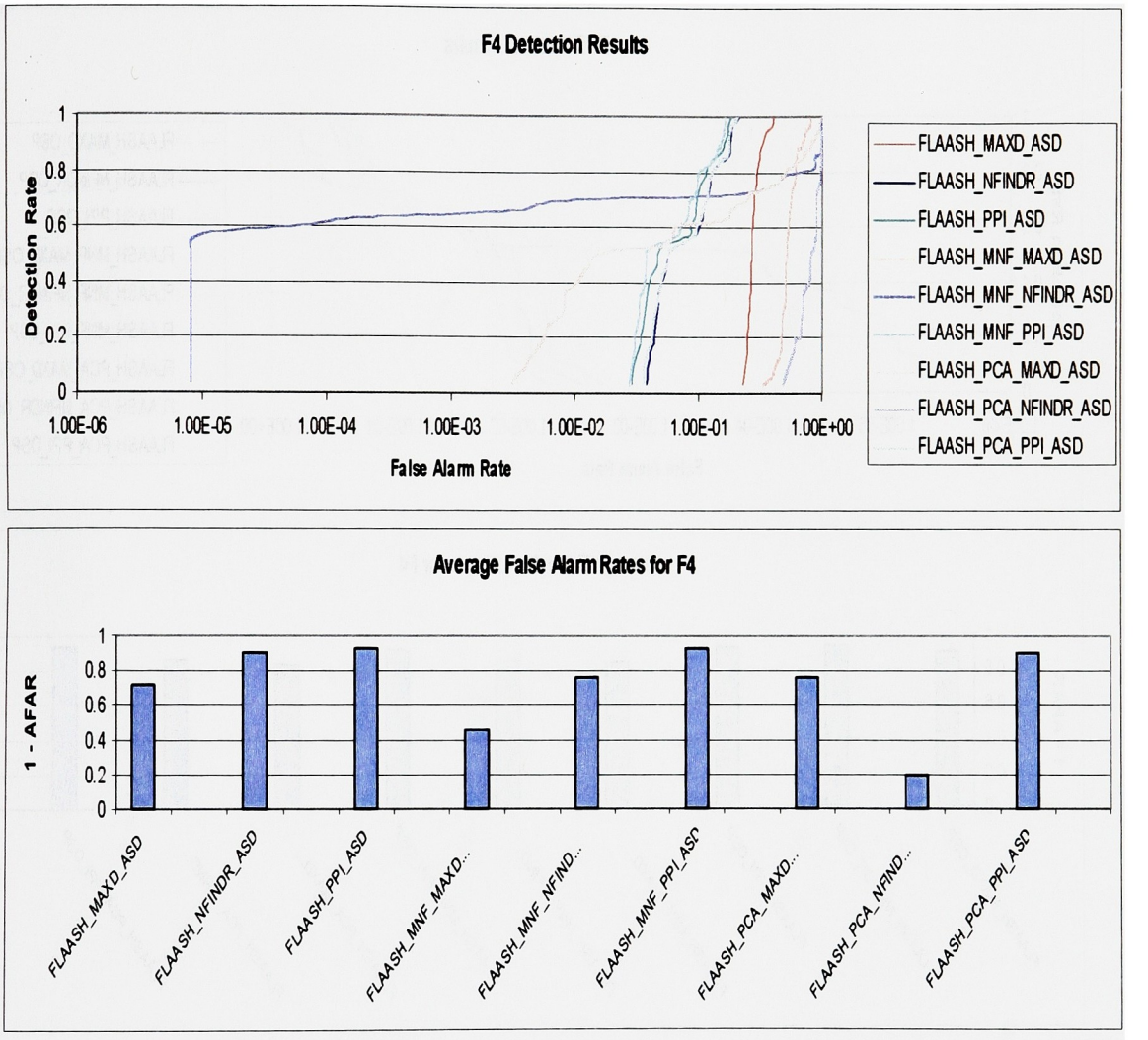

Figure A.17: ROC and (1 − AFAR) graphs for the F4 target that employed the FLAASH atmospheric compensation algorithm and the ASD matched filter.

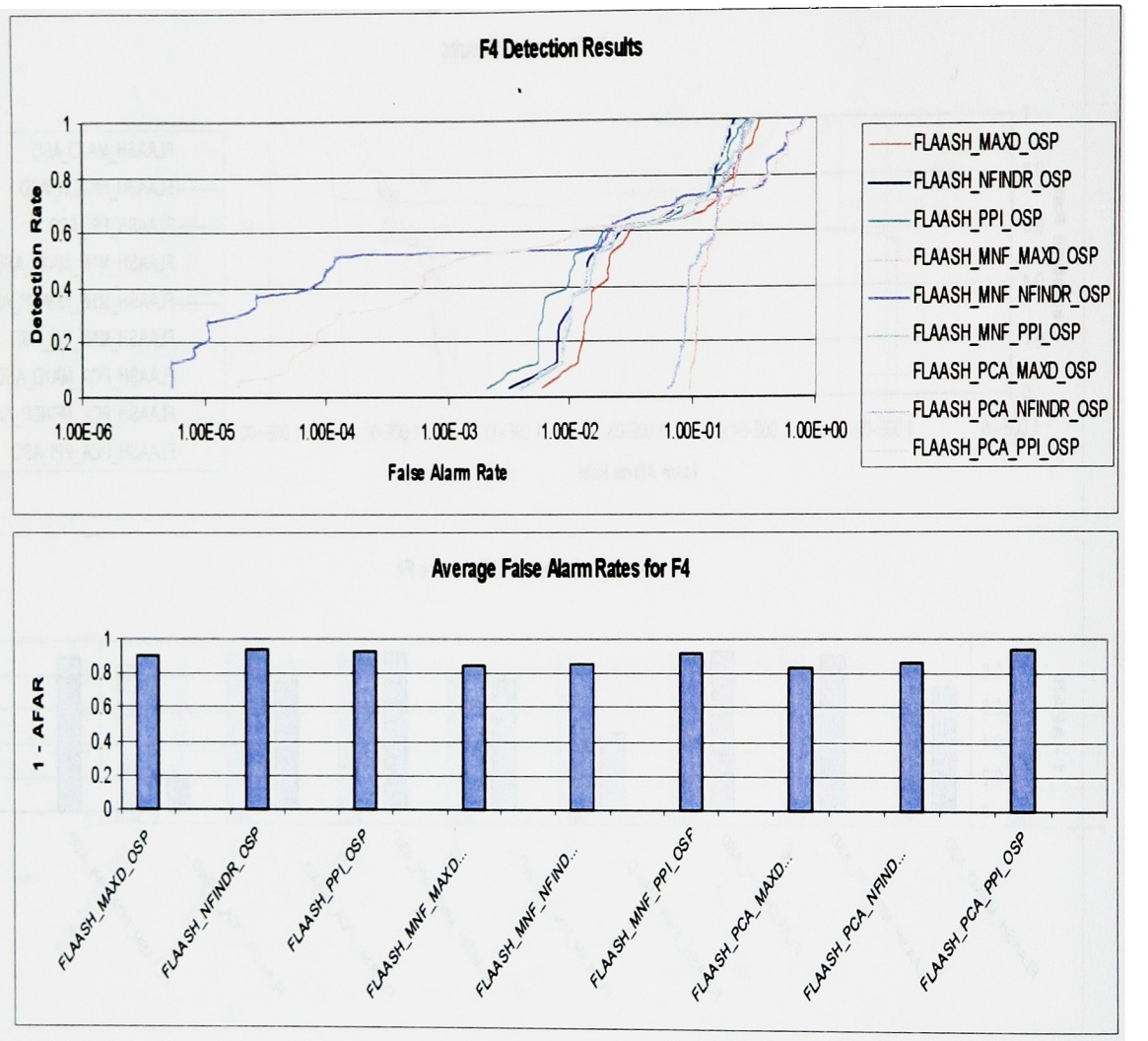

Figure A.18: ROC and (1 − AFAR) graphs for the F4 target that employed the FLAASH atmospheric compensation algorithm and the OSP matched filter.

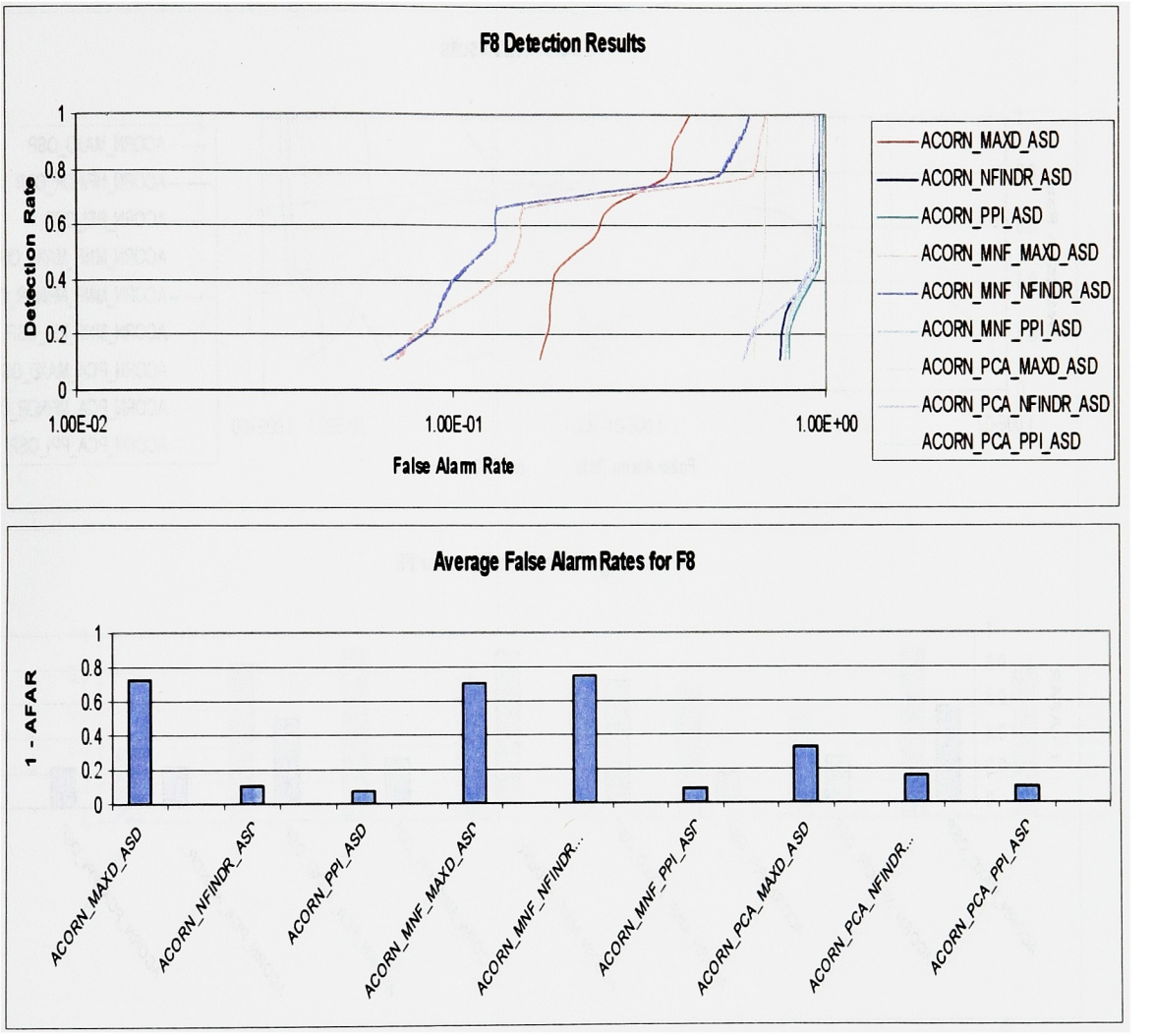

Figure A.19: ROC and (1 – AFAR) graphs for the F8 target that employed the ACORN atmospheric compensation algorithm and the ASD matched filter.

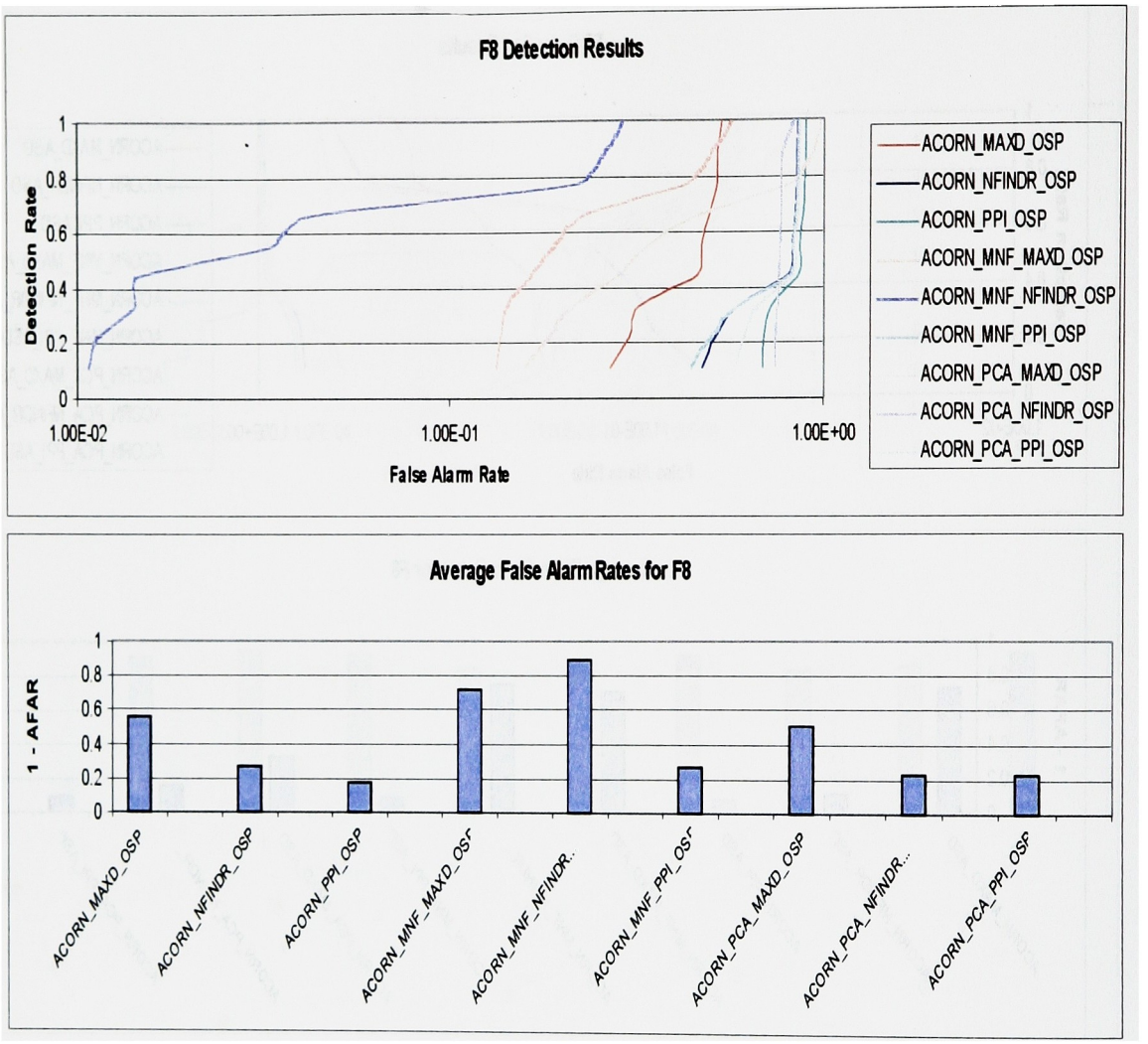

Figure A.20: ROC and (1 − AFAR) graphs for the F8 target that employed the ACORN atmospheric compensation algorithm and the OSP matched filter.

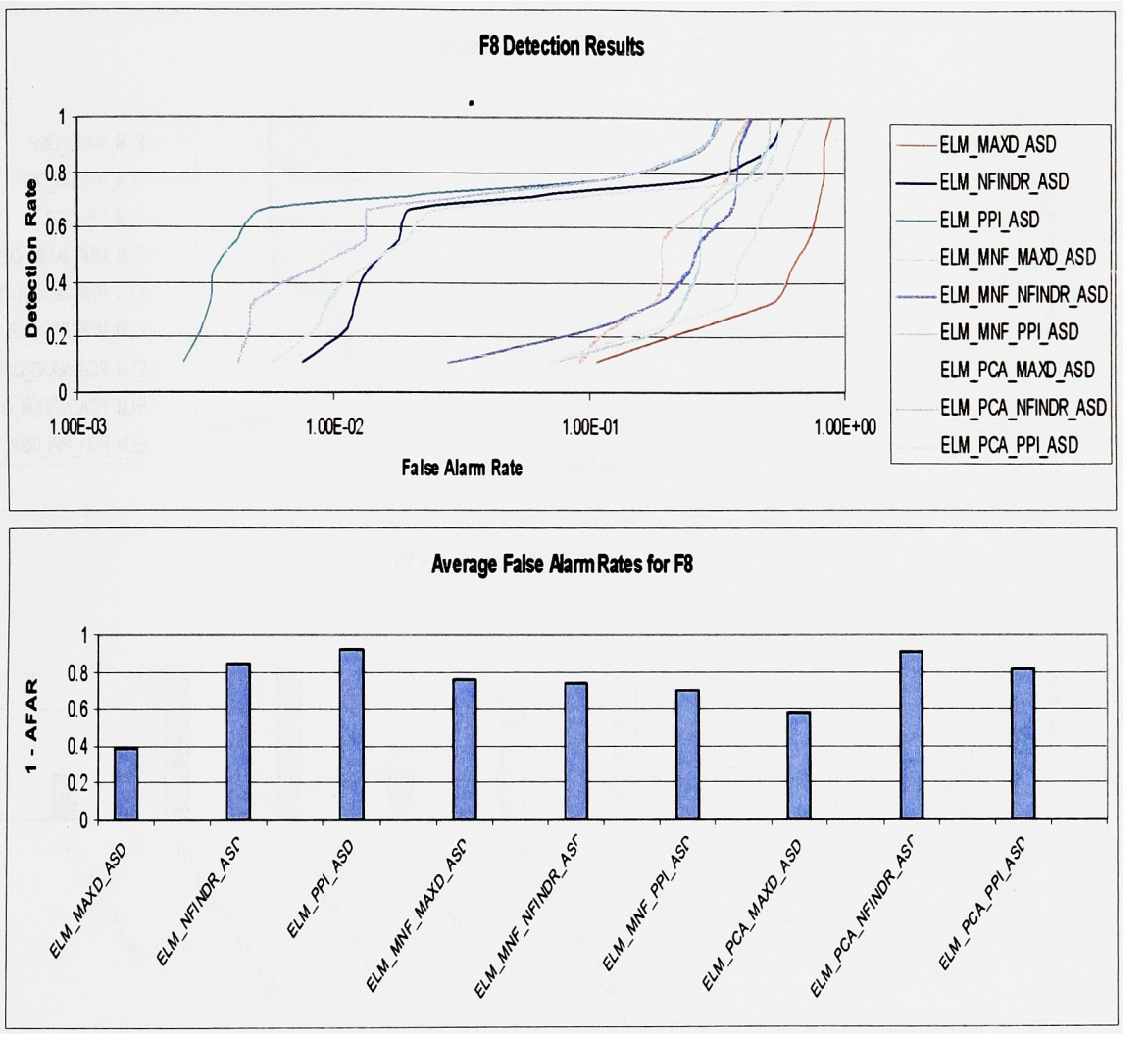

Figure A.21: ROC and (1 – AFAR) graphs for the F8 target that employed the ELM atmospheric compensation algorithm and the ASD matched filter.

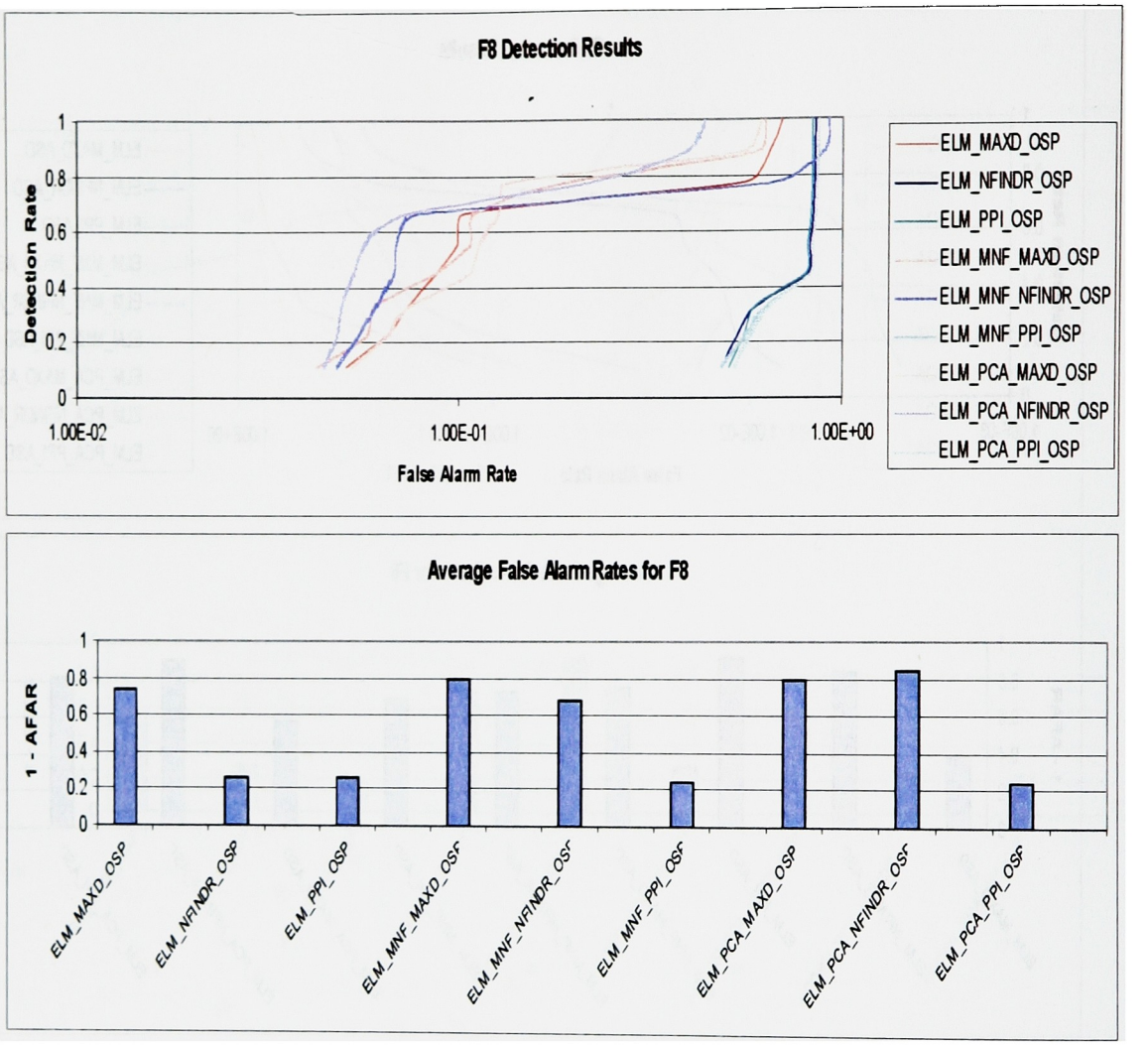

Figure A.22: ROC and (1 − AFAR) graphs for the F8 target that employed the ELM atmospheric compensation algorithm and the OSP matched filter.

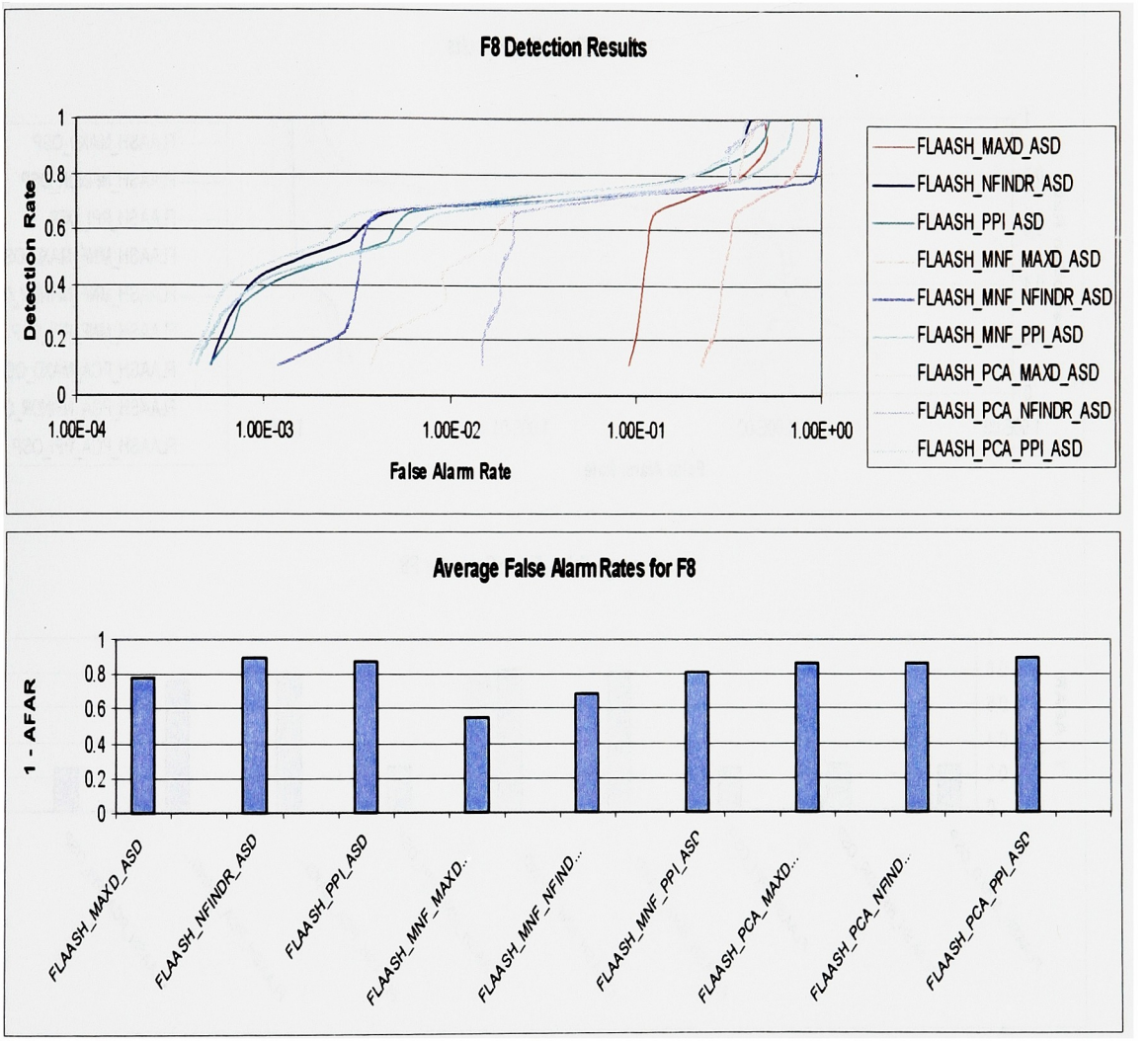

Figure A.23: ROC and (1 – AFAR) graphs for the F8 target that employed the FLAASH atmospheric compensation algorithm and the ASD matched filter.

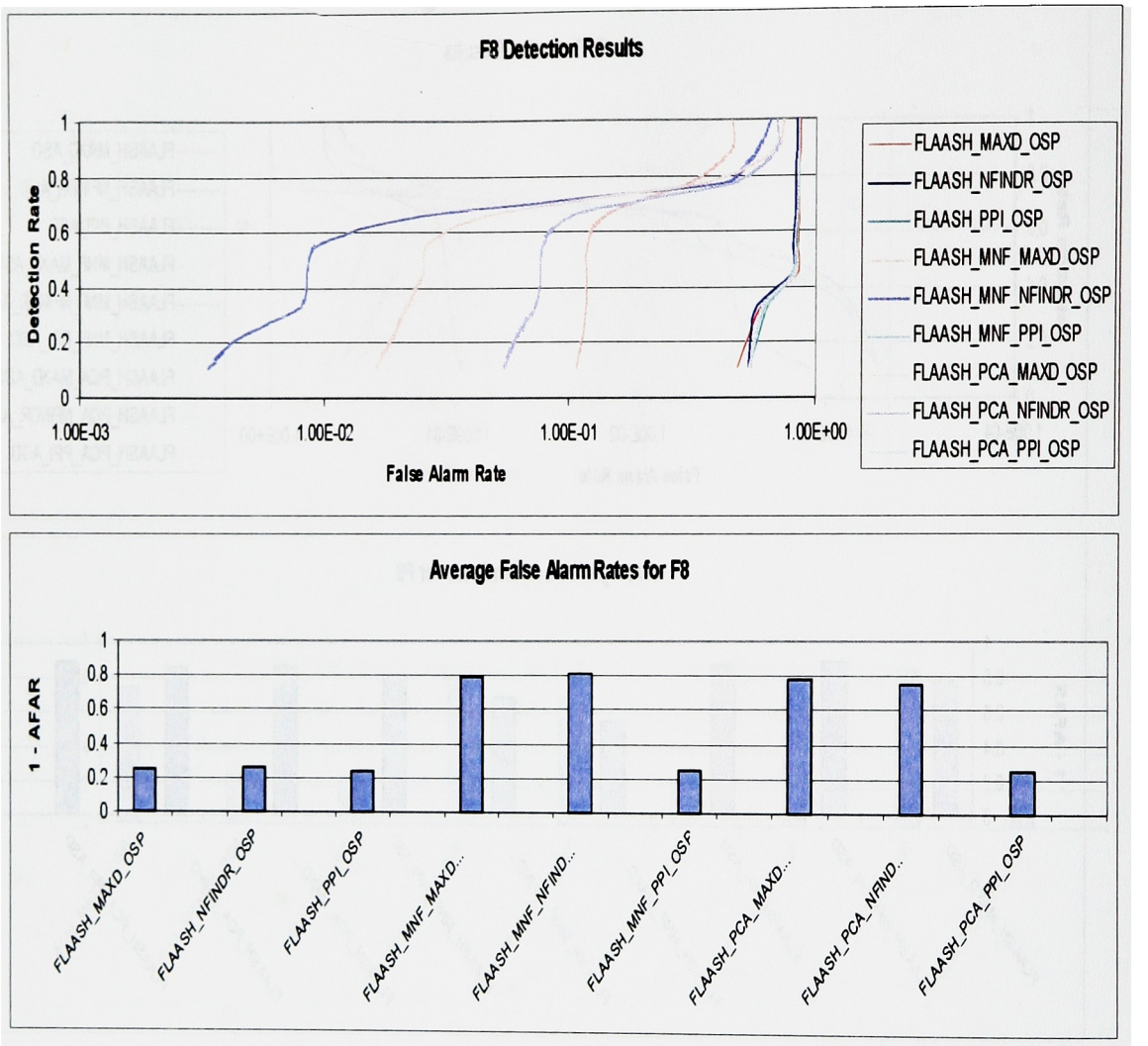

Figure A.24: ROC and (1 – AFAR) graphs for the F8 target that employed the FLAASH atmospheric compensation algorithm and the OSP matched filter.

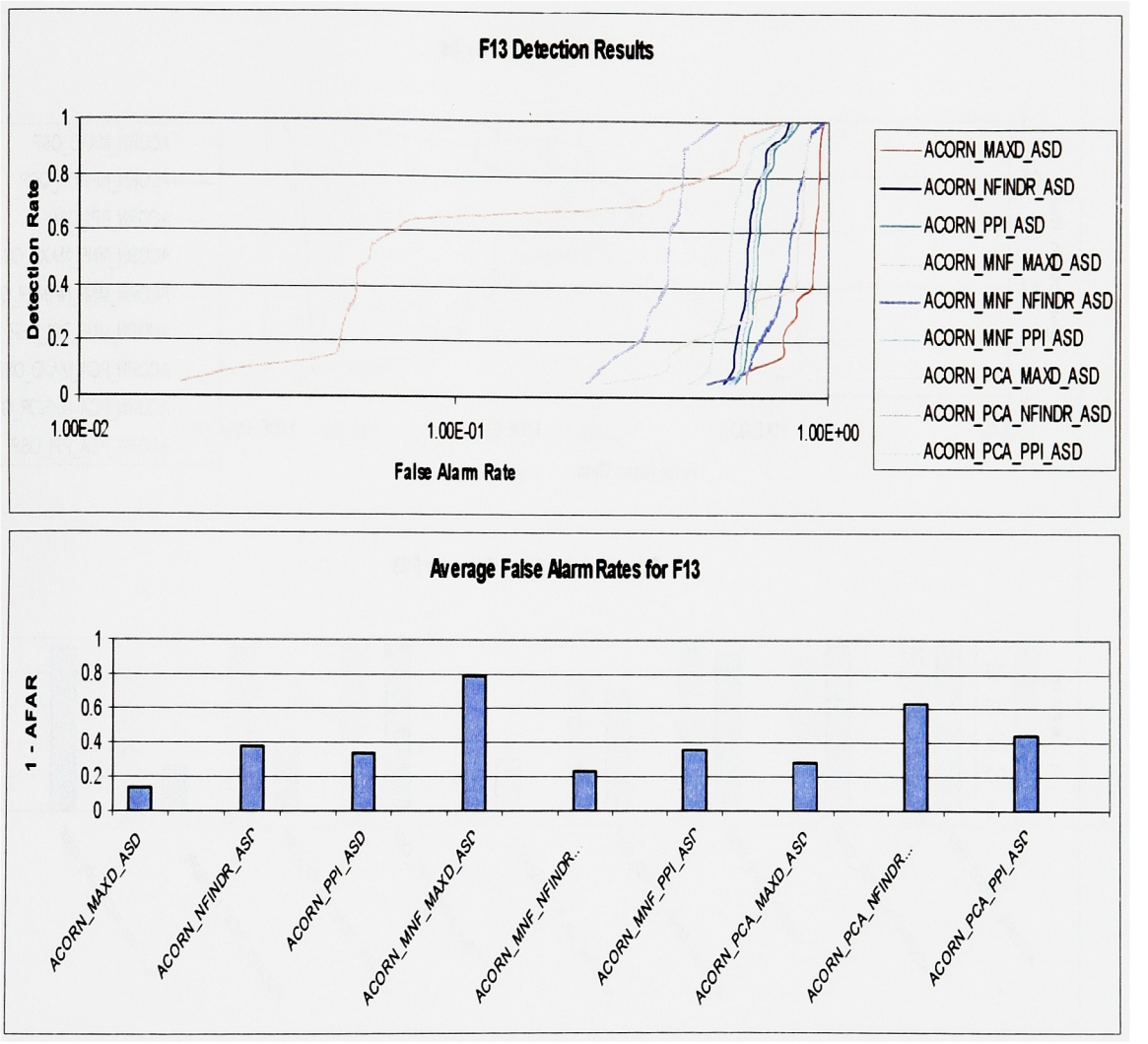

Figure A.25: ROC and (1 – AFAR) graphs for the F13 target that employed the ACORN atmospheric compensation algorithm and the ASD matched filter.

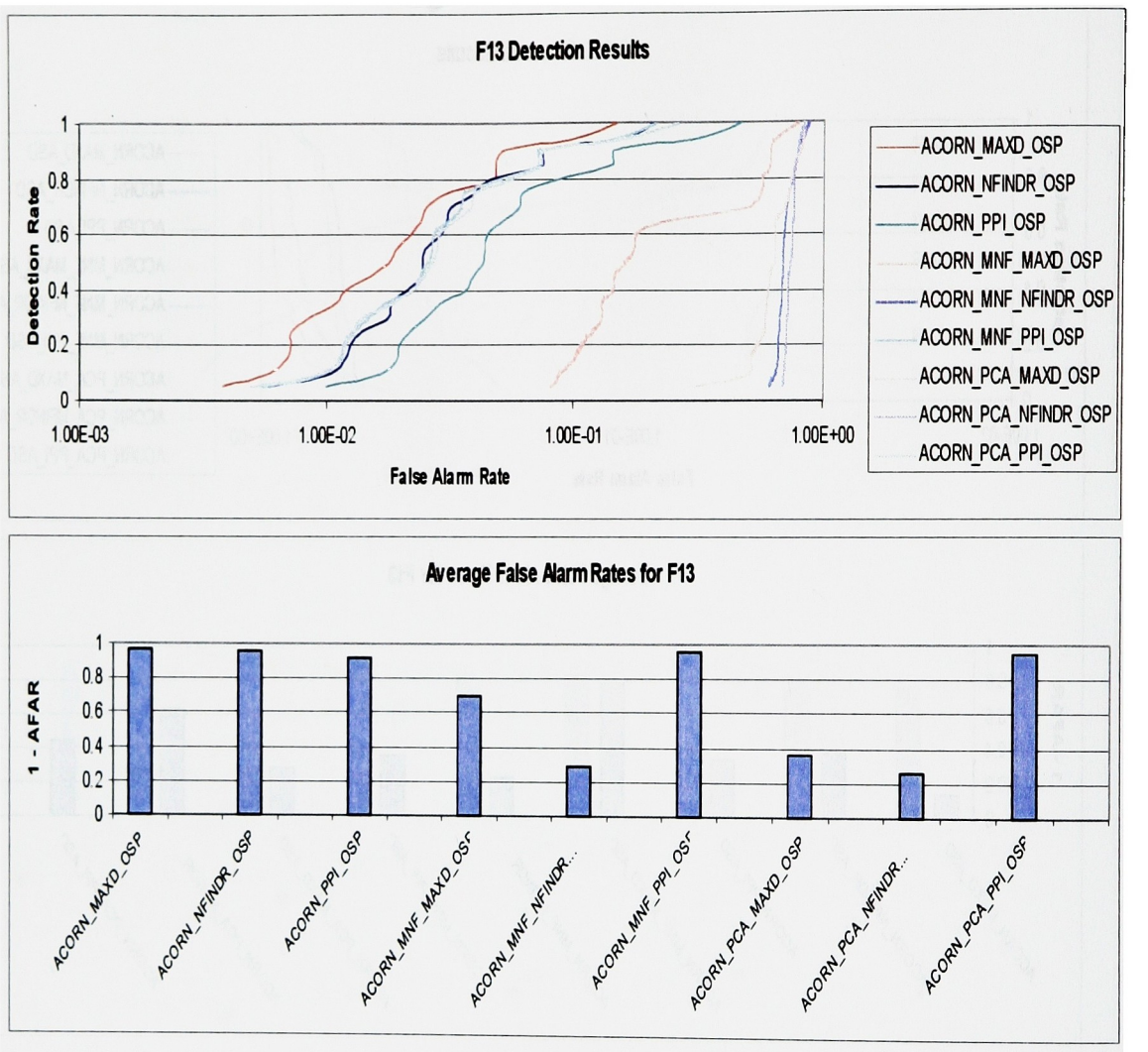

Figure A.26: ROC and (1 – AFAR) graphs for the F13 target that employed the ACORN atmospheric compensation algorithm and the OSP matched filter.

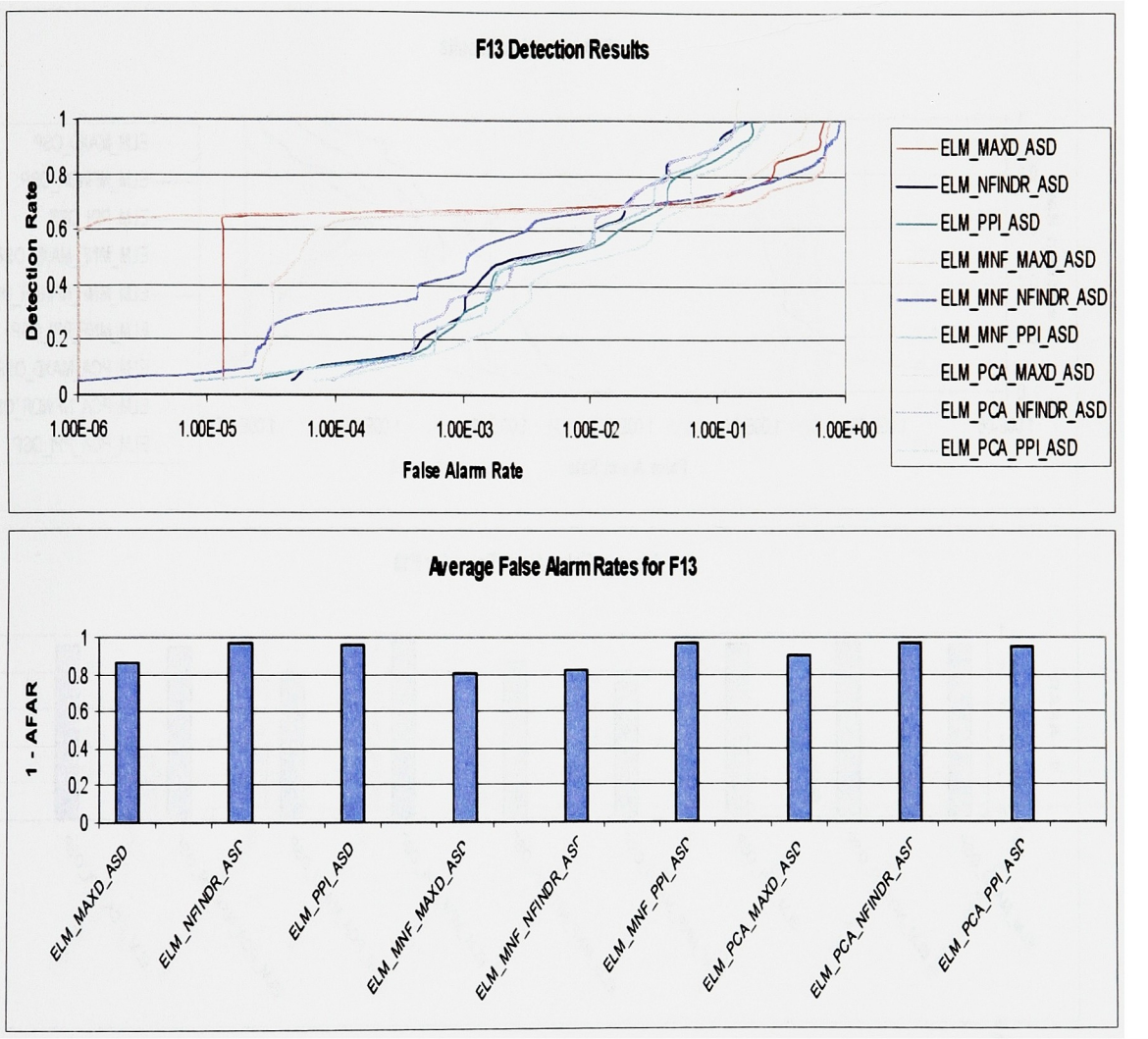

Figure A.27: ROC and (1 – AFAR) graphs for the F13 target that employed the ELM atmospheric compensation algorithm and the ASD matched filter.

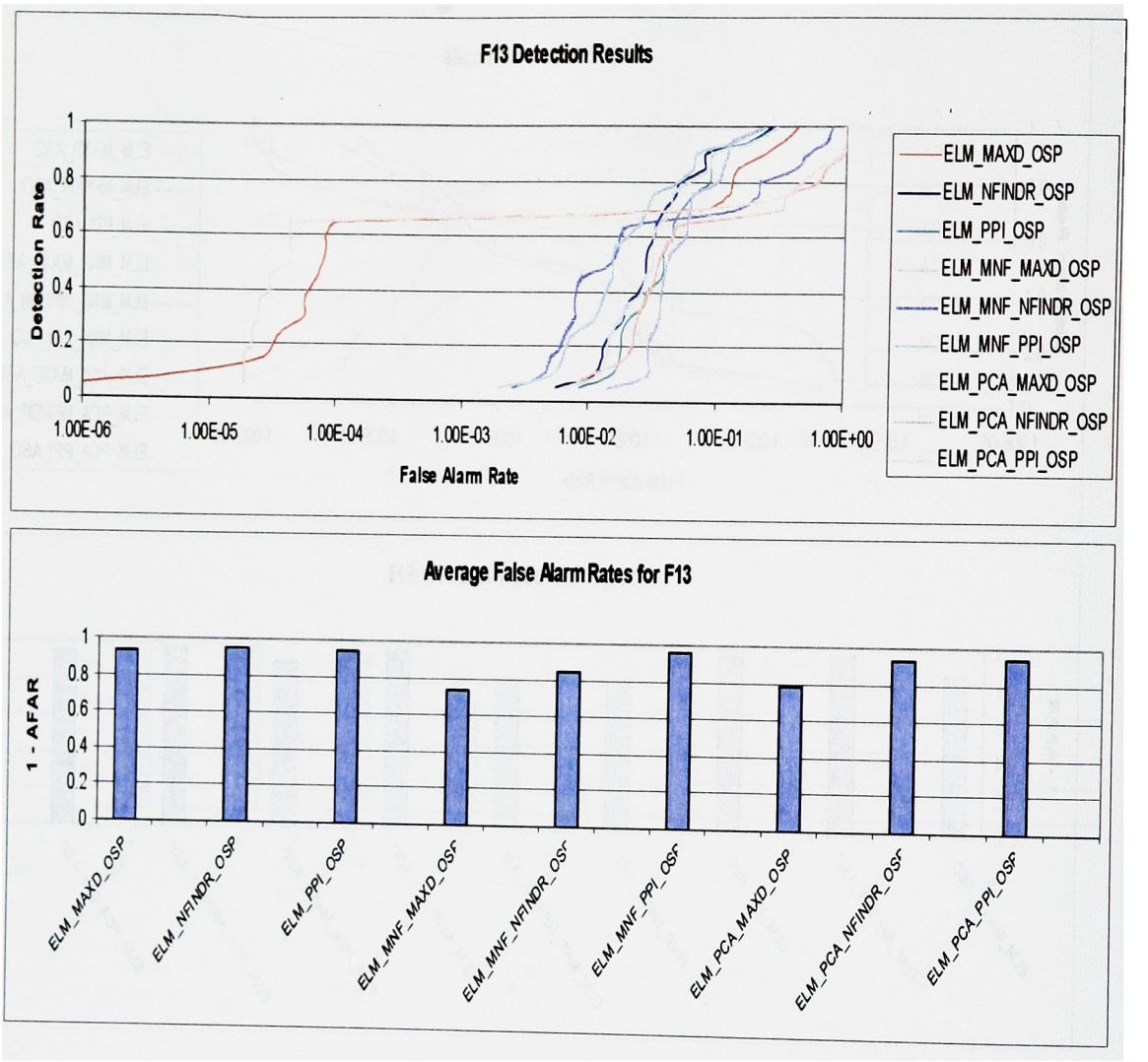

Figure A.28: ROC and (1 – AFAR) graphs for the F13 target that employed the ELM atmospheric compensation algorithm and the OSP matched filter.

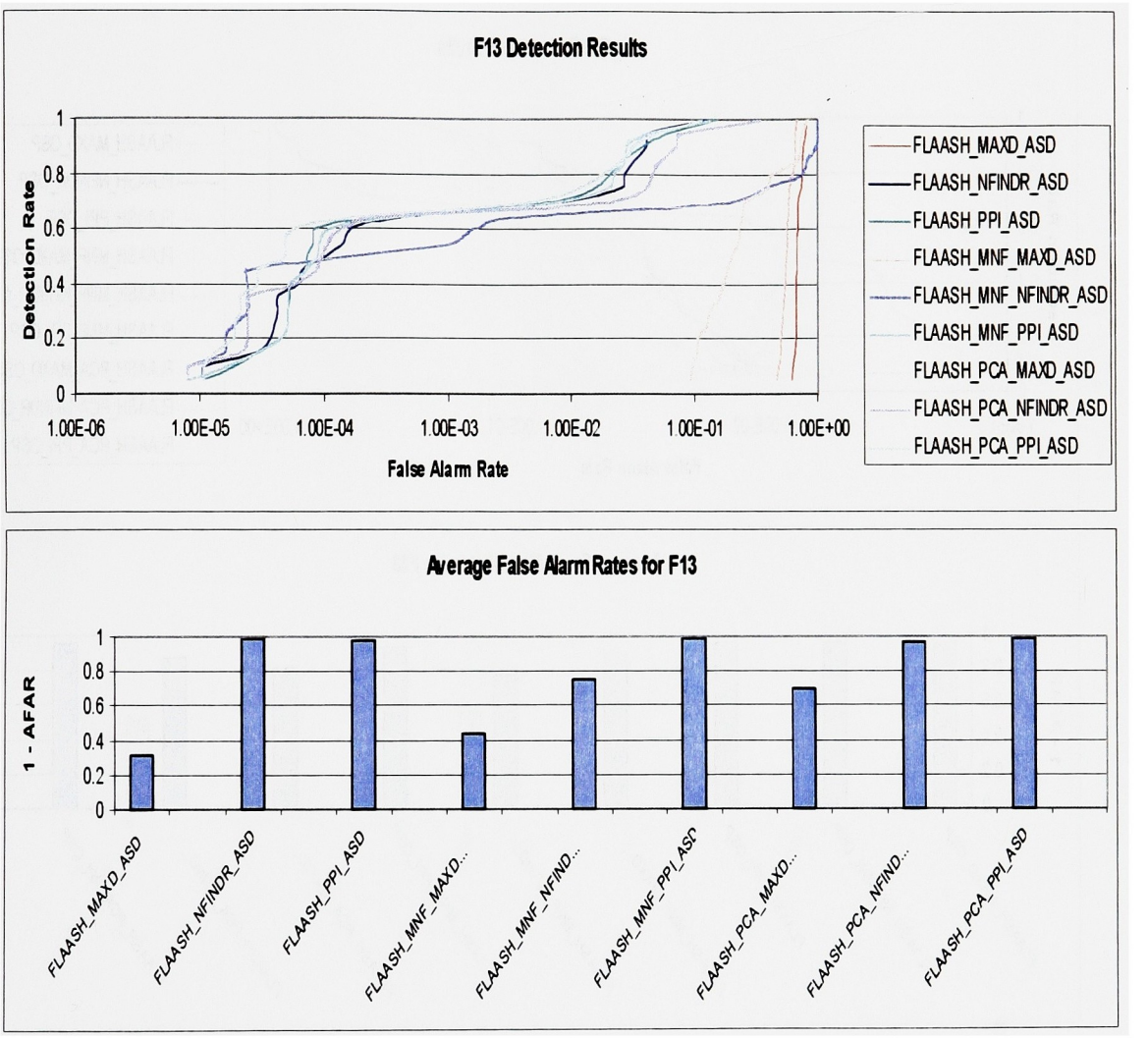

Figure A.29: ROC and (1 − AFAR) graphs for the F13 target that employed the FLAASH atmospheric compensation algorithm and the ASD matched filter.

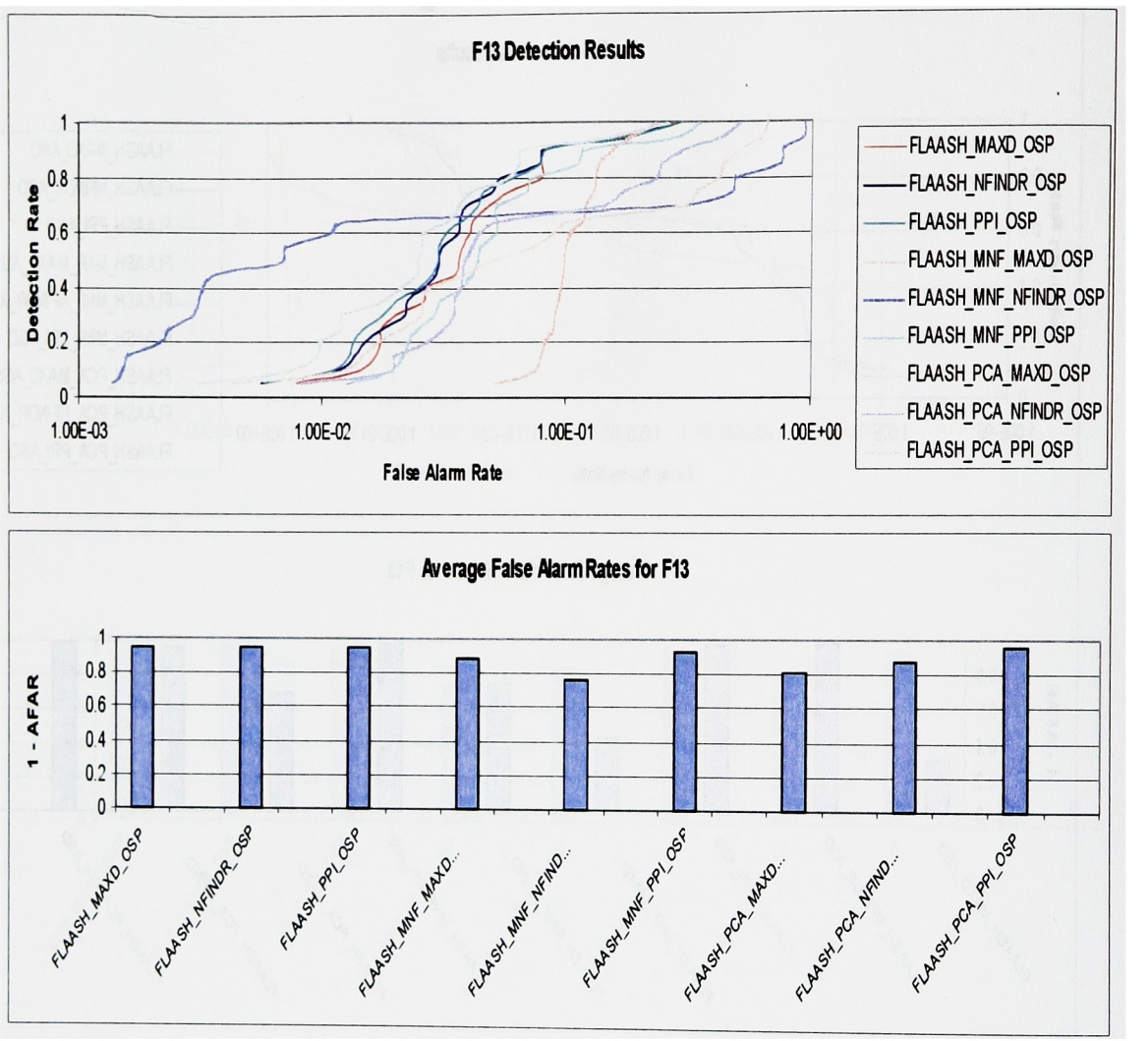

Figure A.30: ROC and (1 − AFAR) graphs for the F13 target that employed the FLAASH atmospheric compensation algorithm and the OSP matched filter.

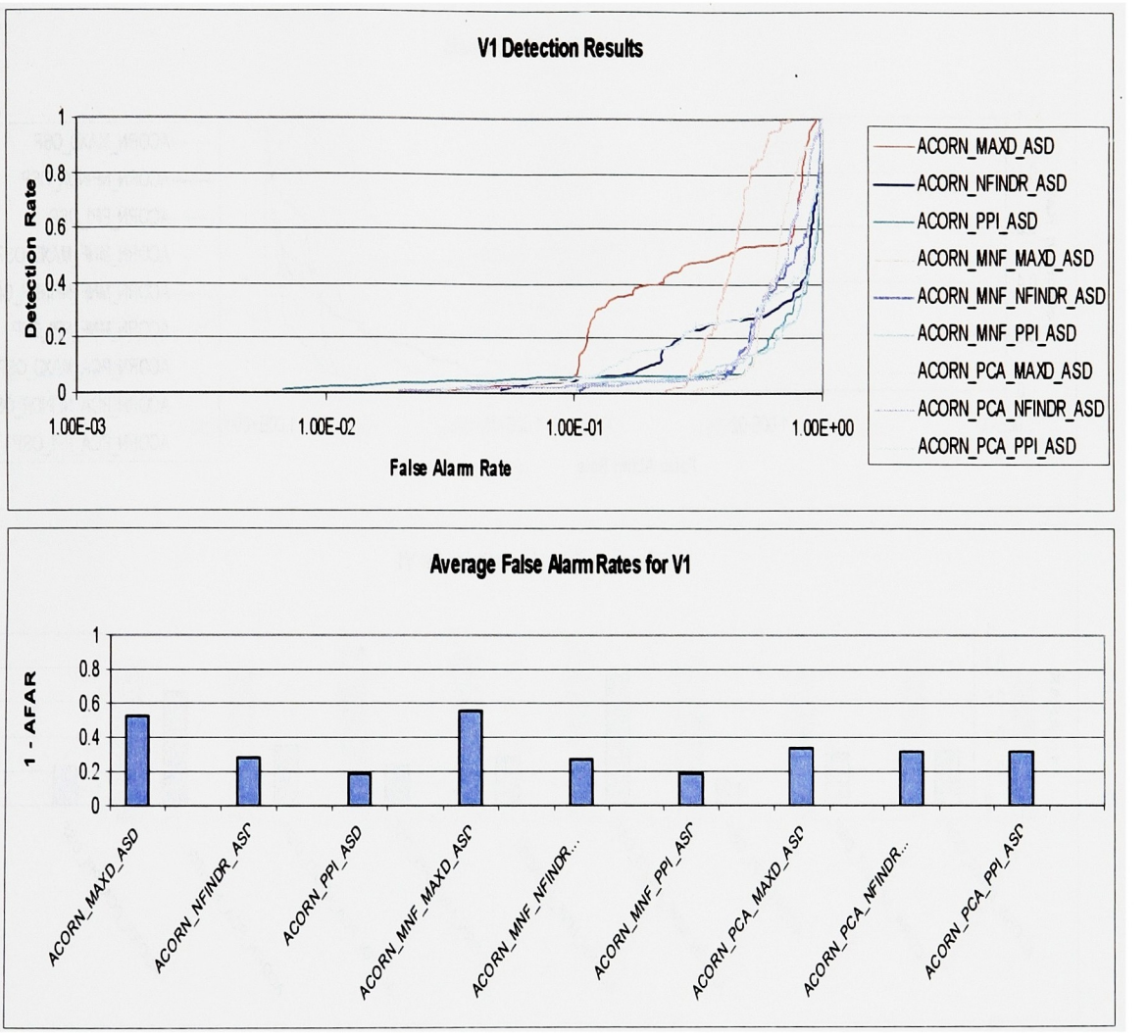

Figure A.31: ROC and (1 – AFAR) graphs for the V1 target that employed the ACORN atmospheric compensation algorithm and the ASD matched filter.

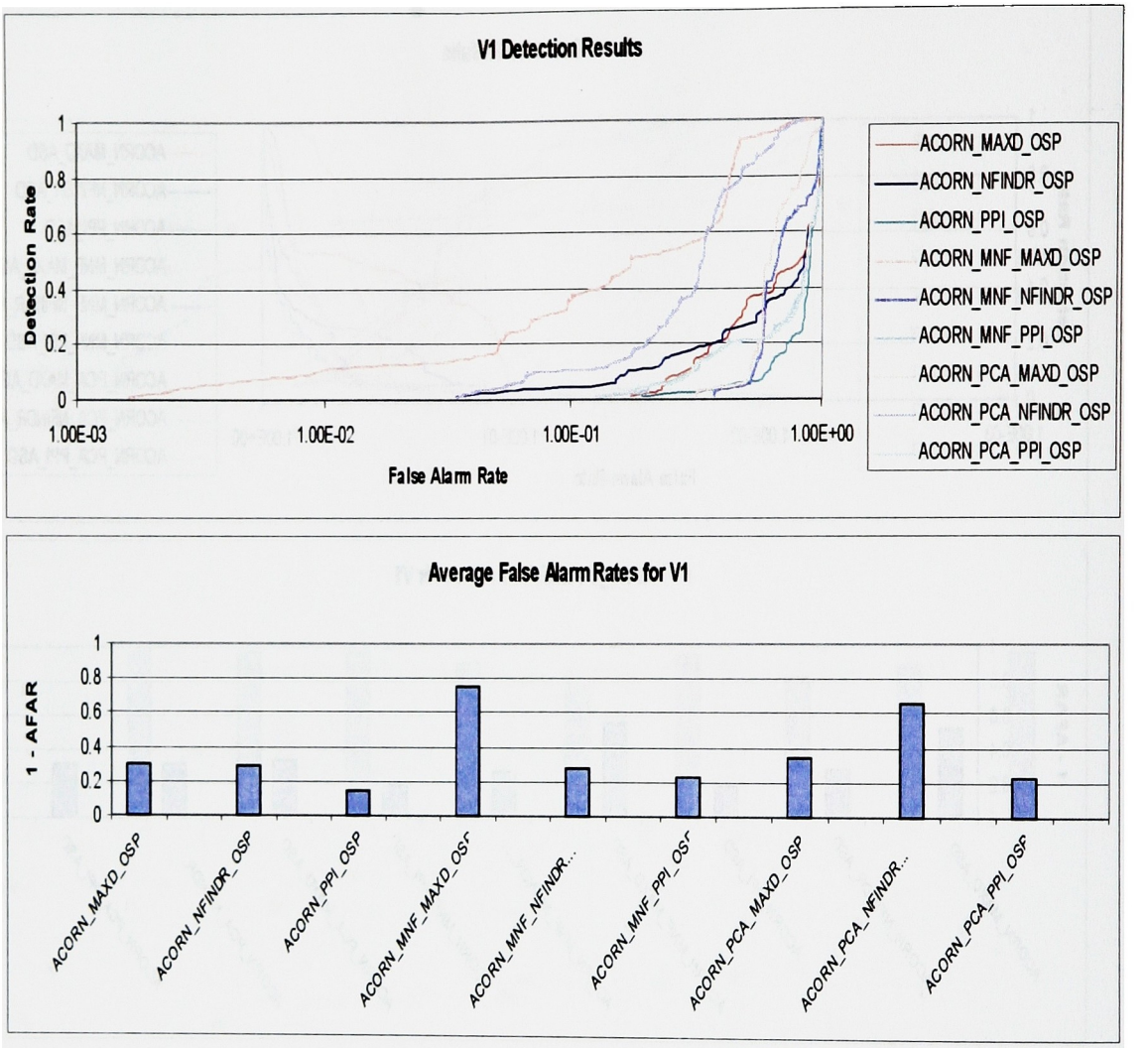

Figure A.32: ROC and (1 – AFAR) graphs for the V1 target that employed the ACORN atmospheric compensation algorithm and the OSP matched filter.

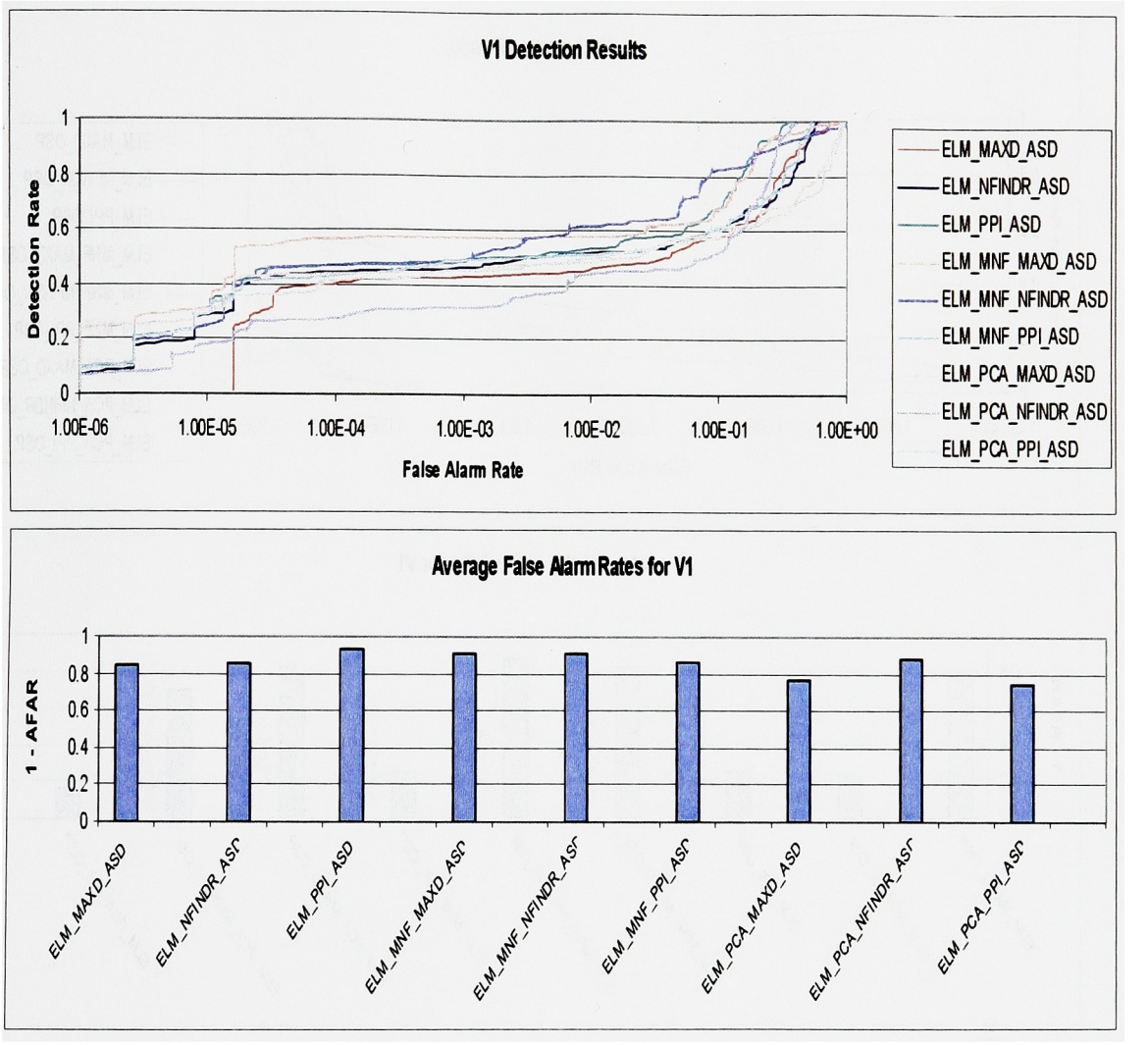

Figure A.33: ROC and (1 – AFAR) graphs for the V1 target that employed the ELM atmospheric compensation algorithm and the ASD matched filter.

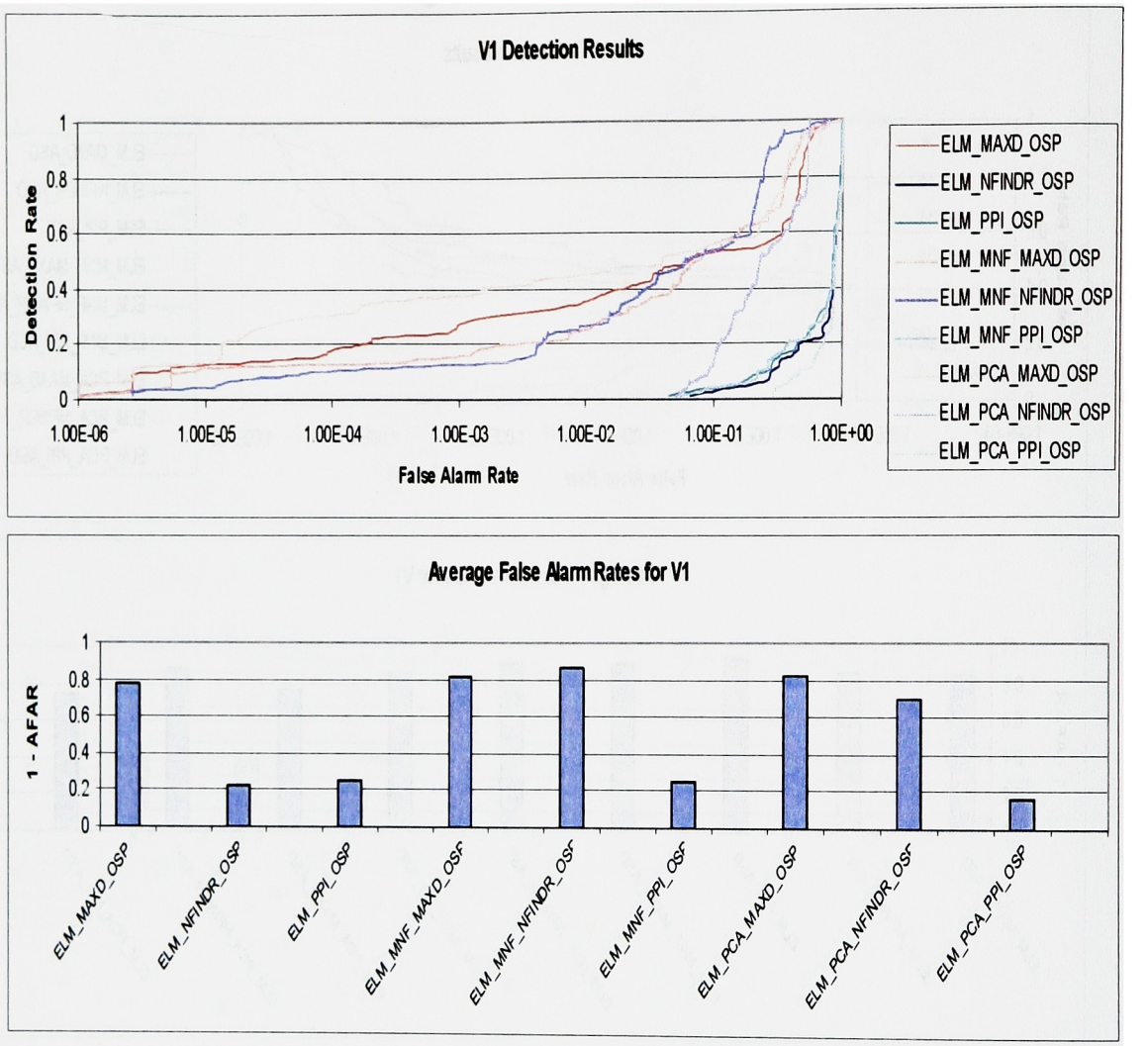

Figure A.34: ROC and (1 − AFAR) graphs for the V1 target that employed the ELM atmospheric compensation algorithm and the OSP matched filter.

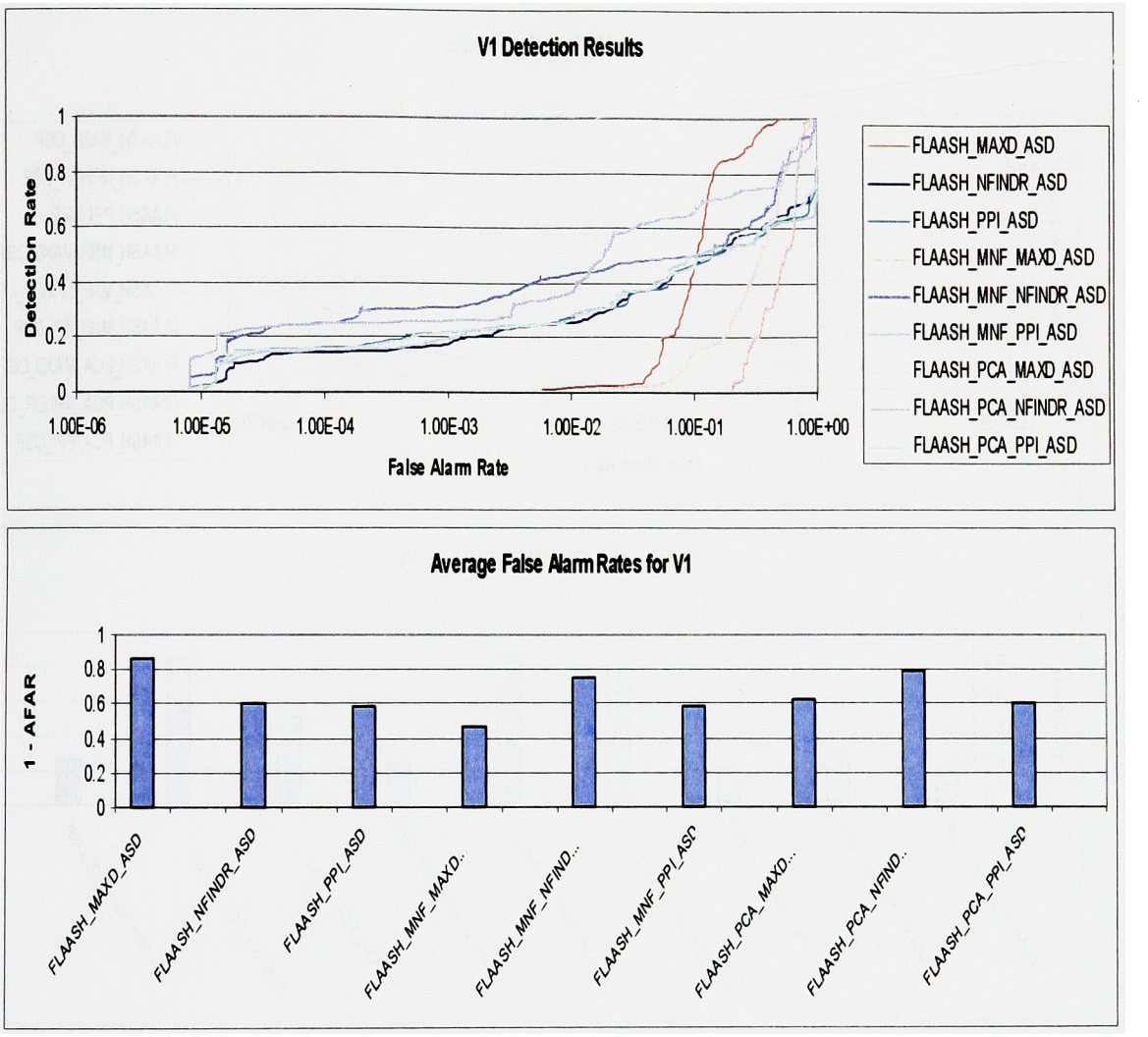

Figure A.35: ROC and (1 – AFAR) graphs for the V1 target that employed the FLAASH atmospheric compensation algorithm and the ASD matched filter.

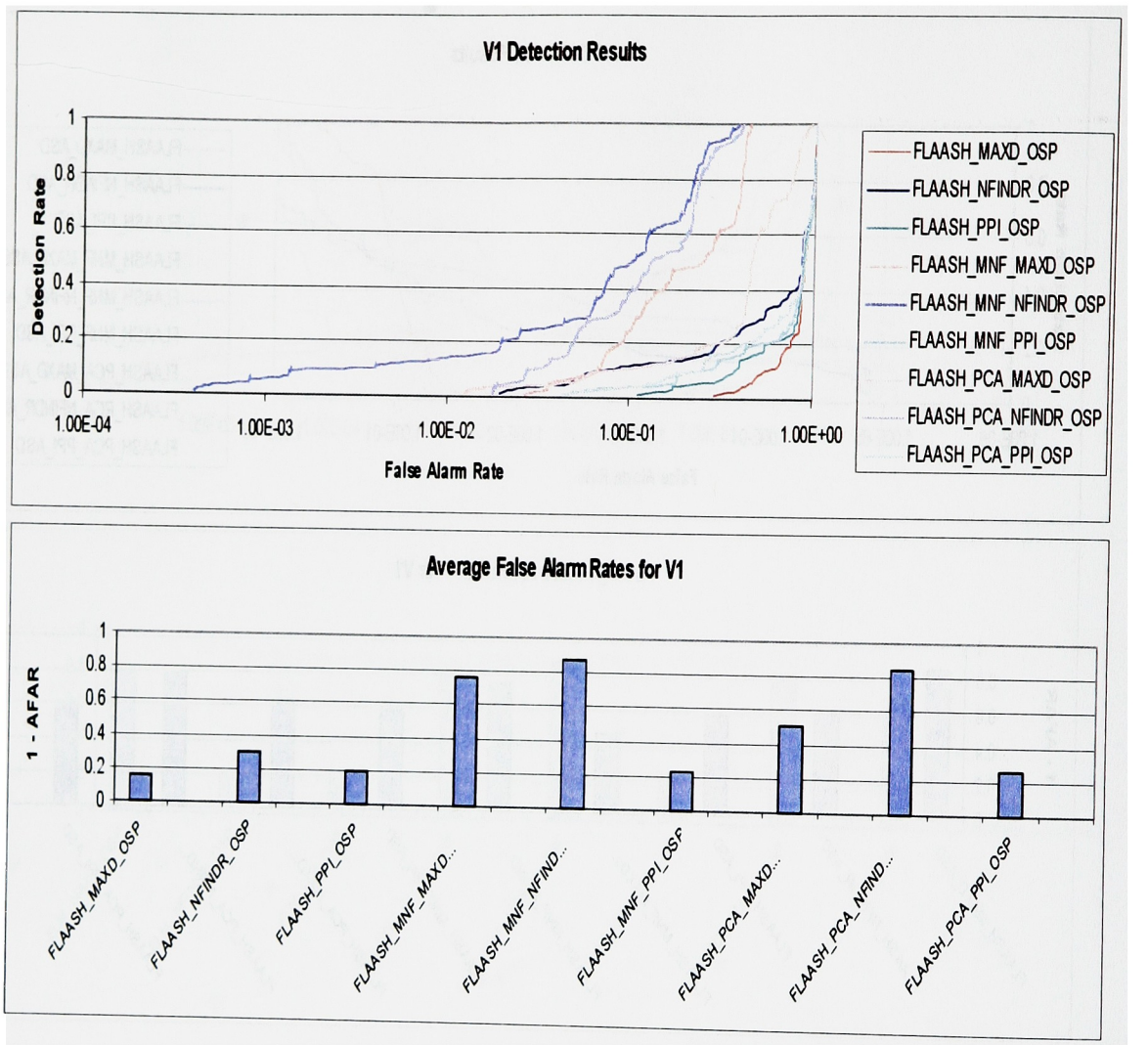

Figure A.36: ROC and (1 – AFAR) graphs for the V1 target that employed the FLAASH atmospheric compensation algorithm and the OSP matched filter.

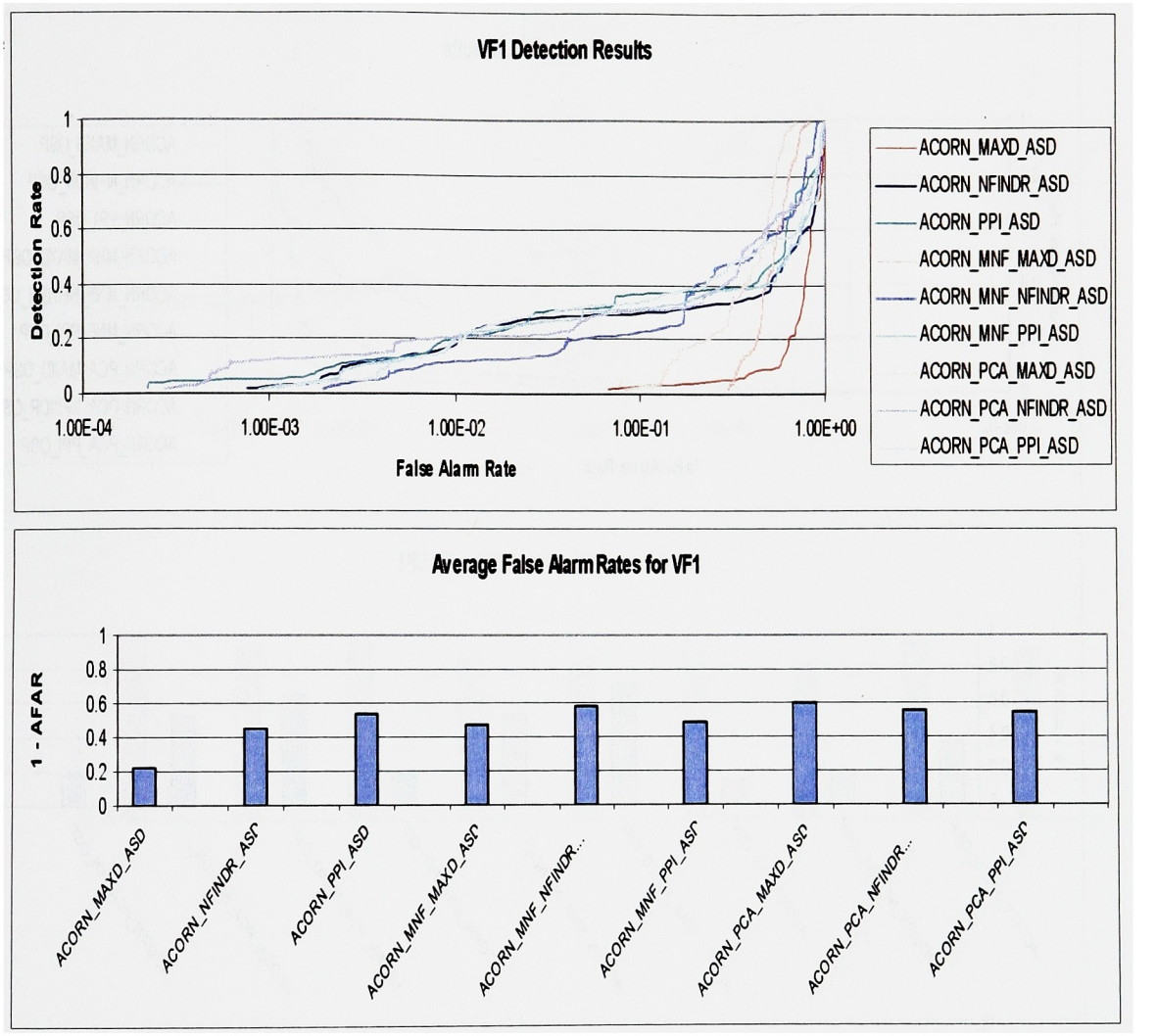

Figure A.37: ROC and (1 – AFAR) graphs for the VF1 target that employed the ACORN atmospheric compensation algorithm and the ASD matched filter.

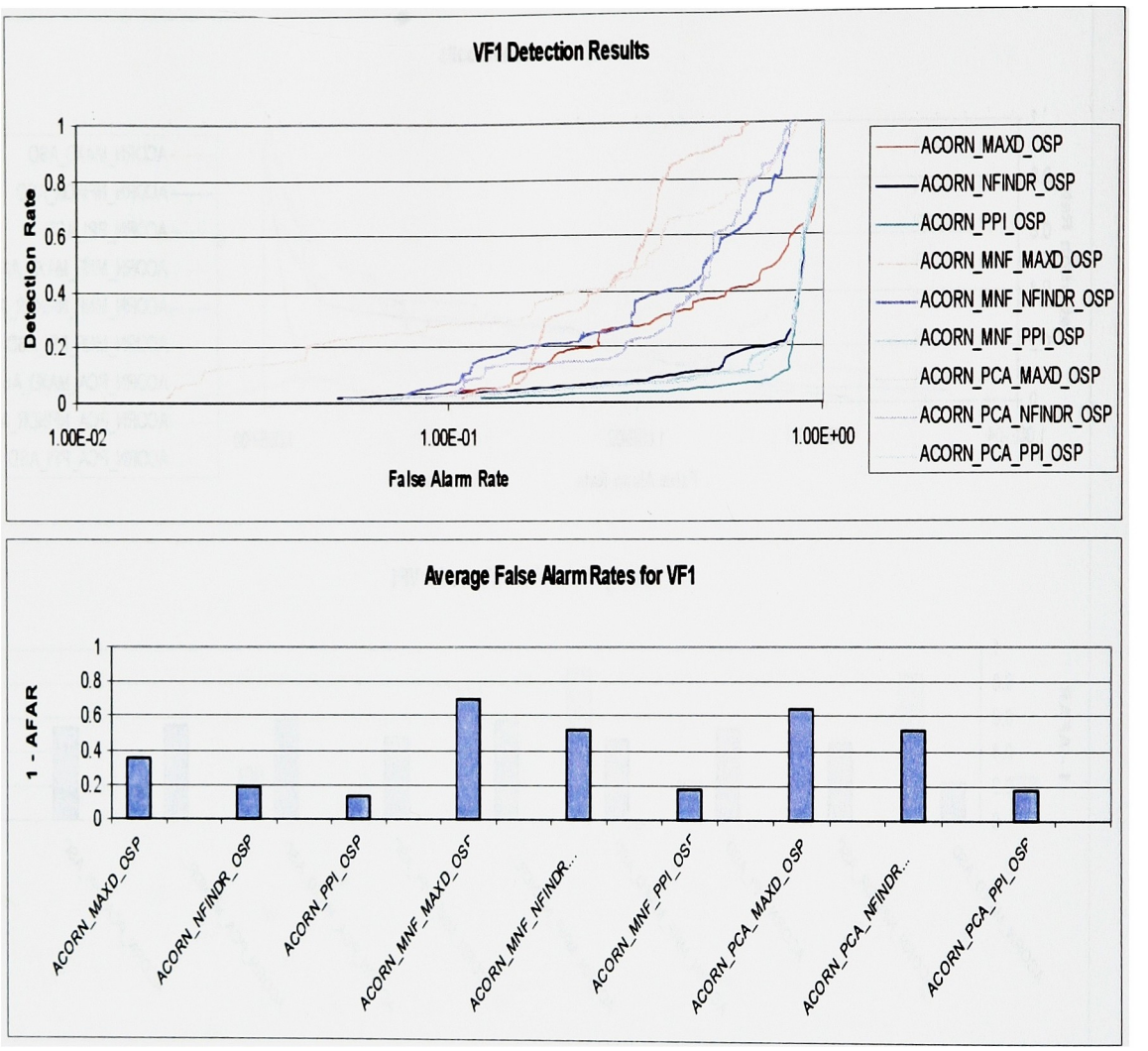

Figure A.38: ROC and (1 − AFAR) graphs for the VF1 target that employed the ACORN atmospheric compensation algorithm and the OSP matched filter.

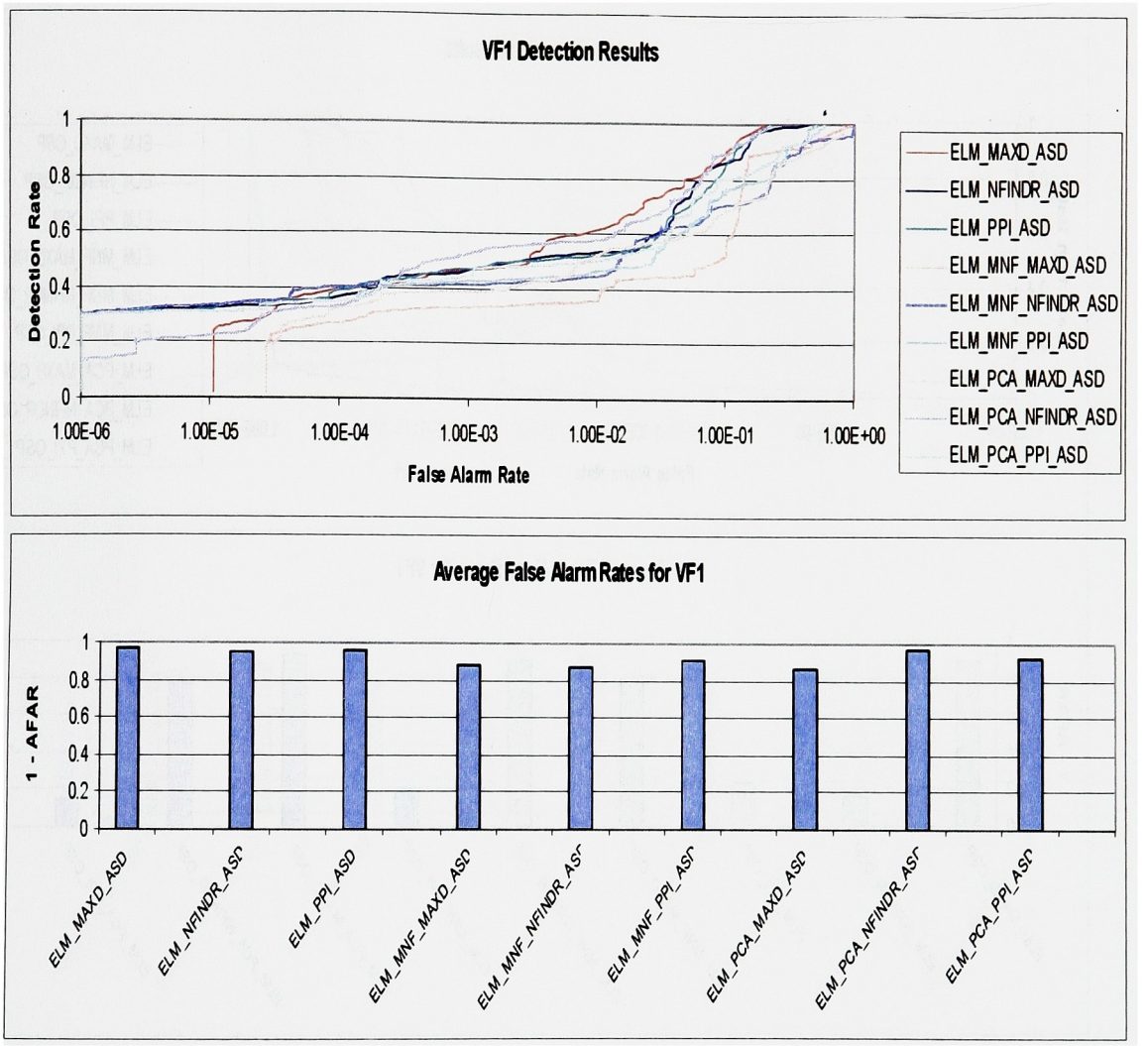

Figure A.39: ROC and (1 – AFAR) graphs for the VF1 target that employed the ELM atmospheric compensation algorithm and the ASD matched filter.

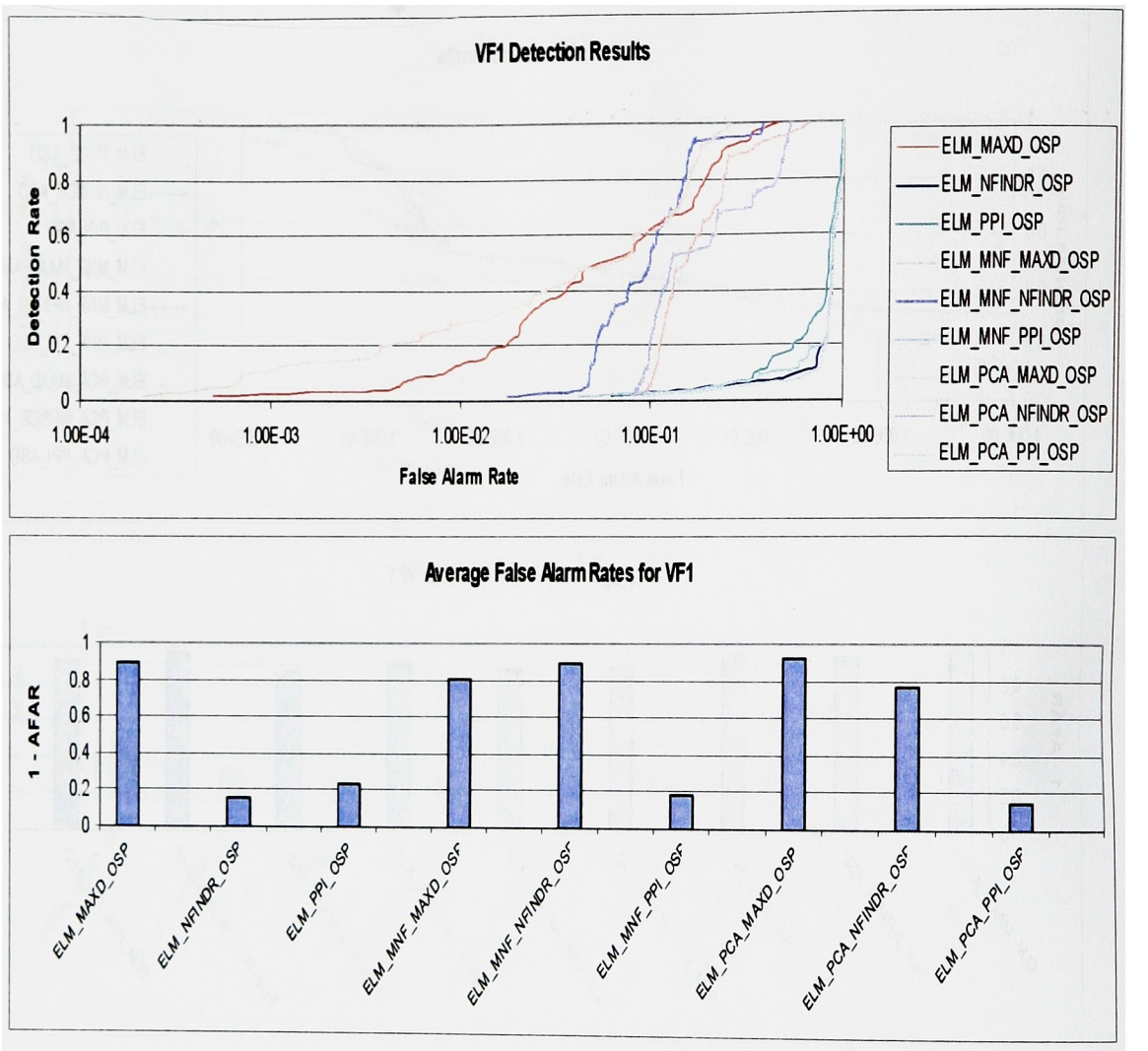

Figure A.40: ROC and (1 – AFAR) graphs for the VF1 target that employed the ELM atmospheric compensation algorithm and the OSP matched filter.

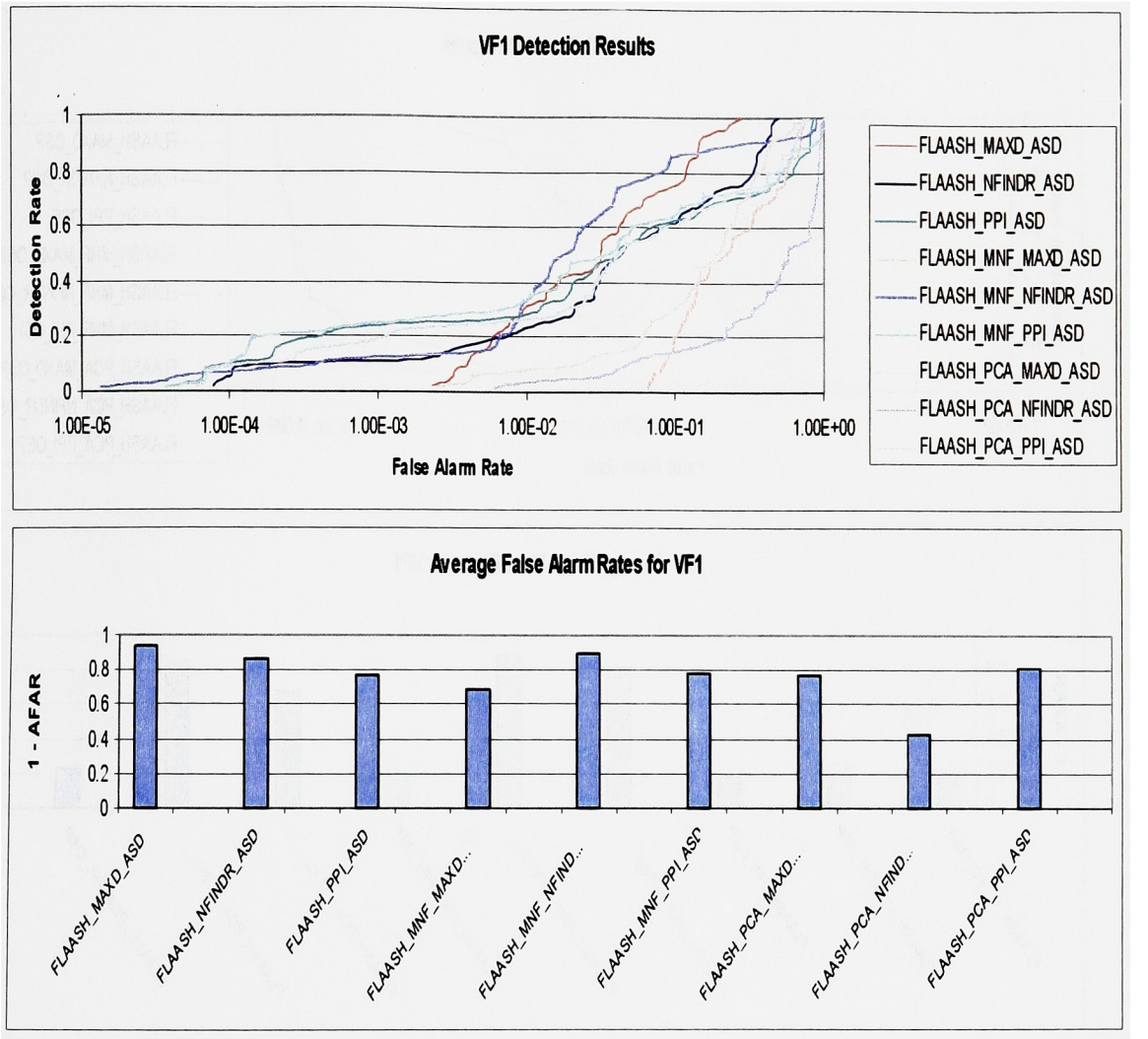

Figure A.41: ROC and (1 – AFAR) graphs for the VF1 target that employed the FLAASH atmospheric compensation algorithm and the ASD matched filter.

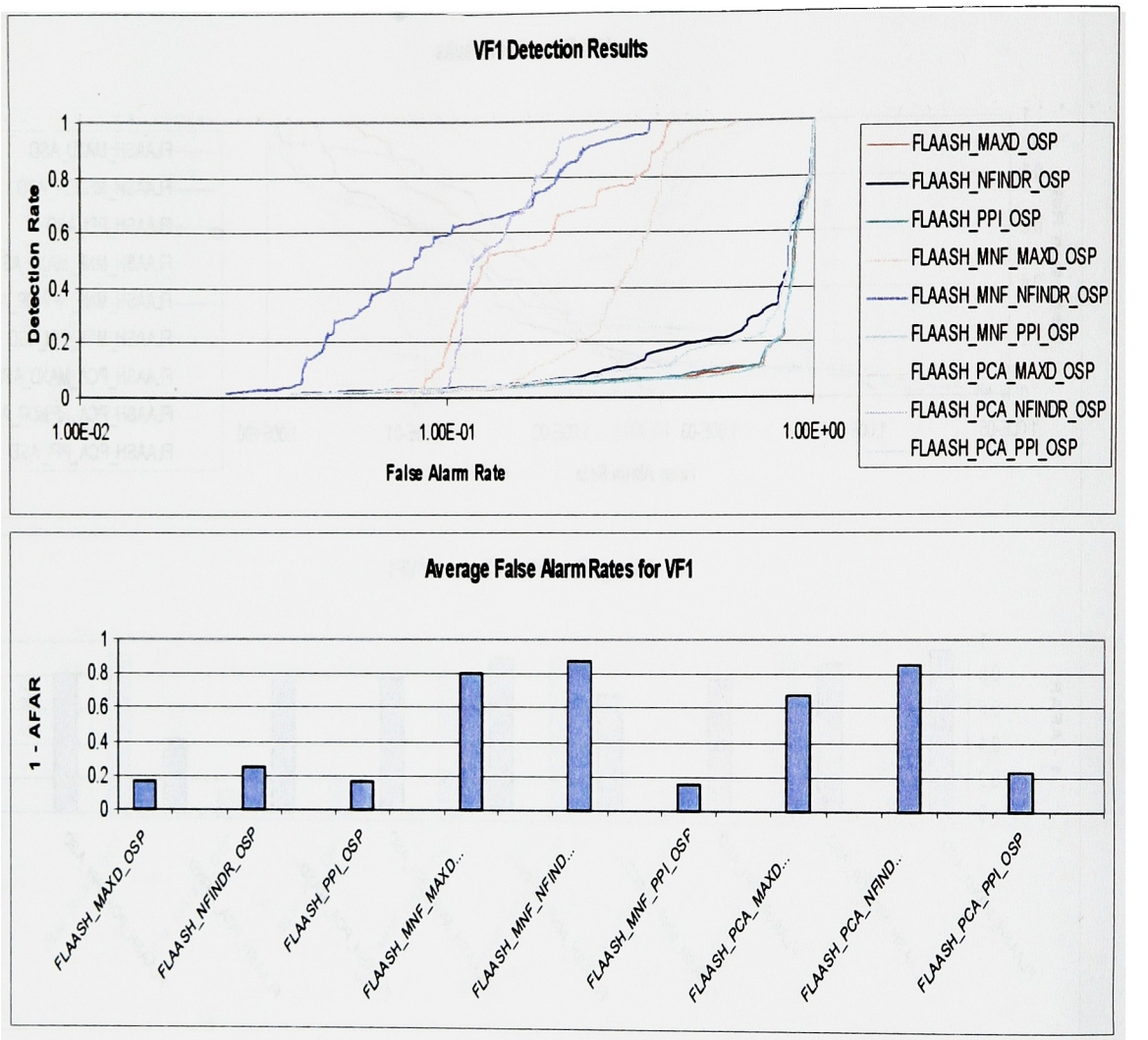

Figure A.42: ROC and (1 – AFAR) graphs for the VF1 target that employed the FLAASH atmospheric compensation algorithm and the OSP matched filter.

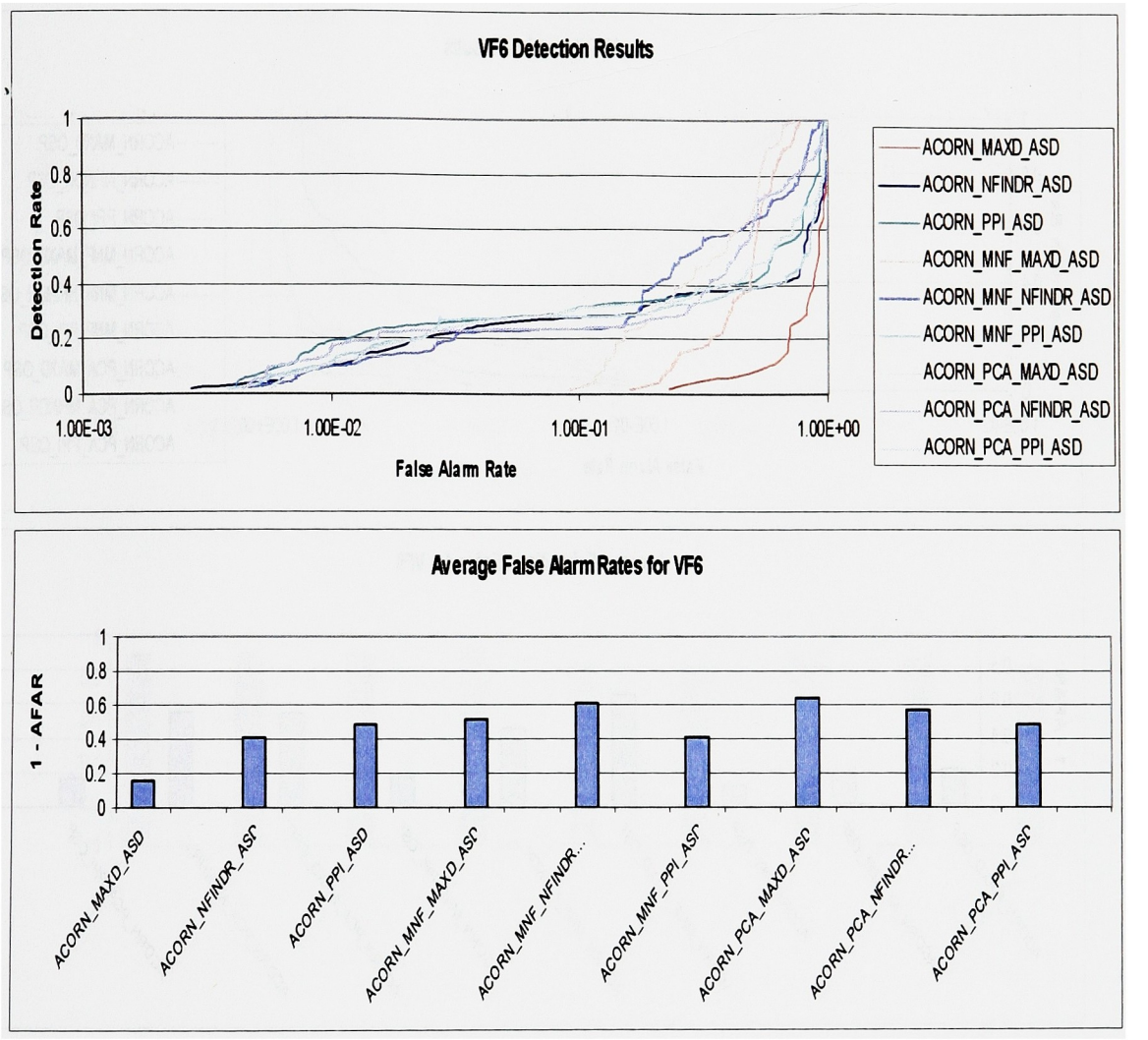

Figure A.43: ROC and (1 − AFAR) graphs for the VF6 target that employed the ACORN atmospheric compensation algorithm and the ASD matched filter.

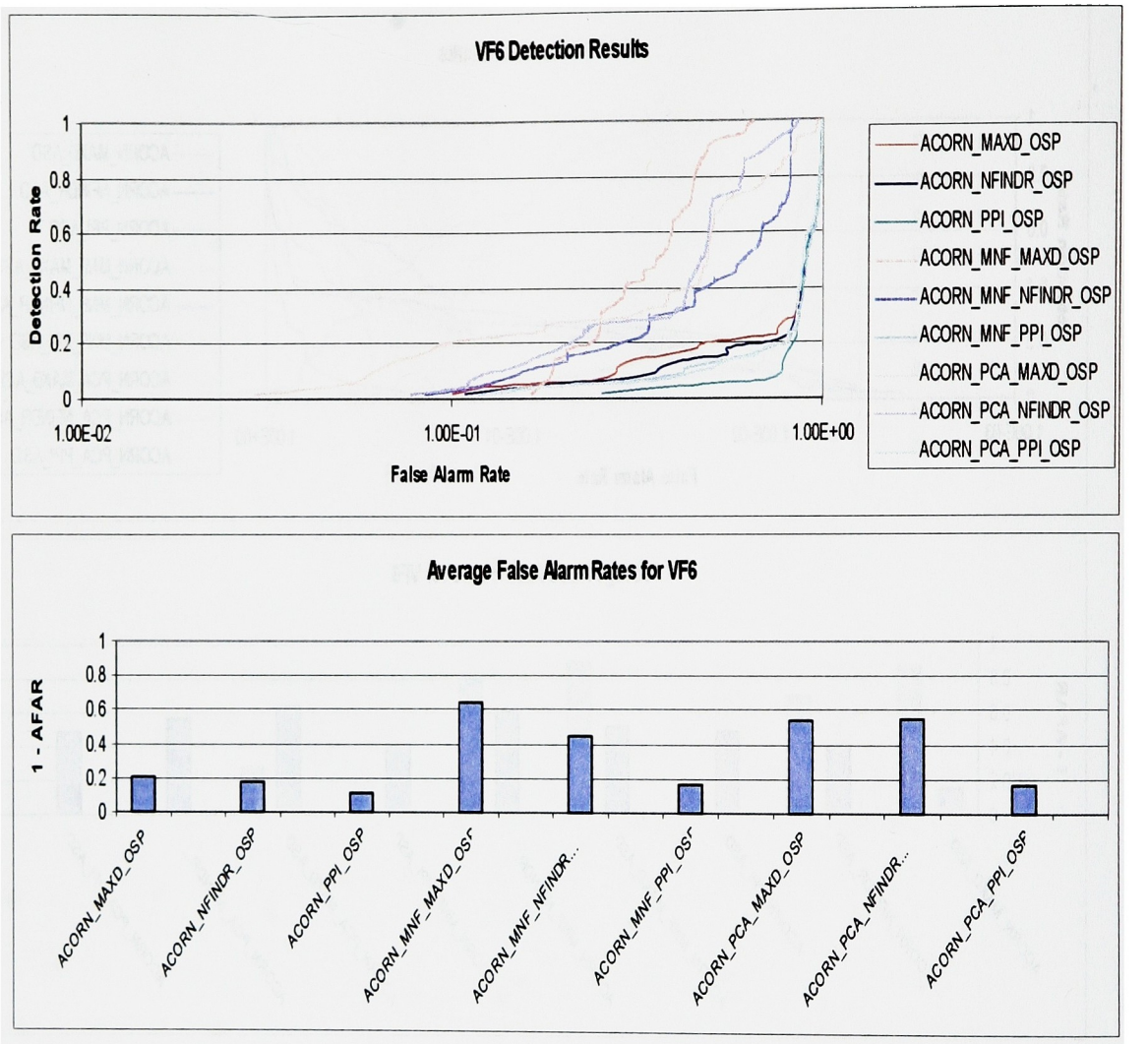

Figure A.44: ROC and (1 − AFAR) graphs for the VF6 target that employed the ACORN atmospheric compensation algorithm and the OSP matched filter.

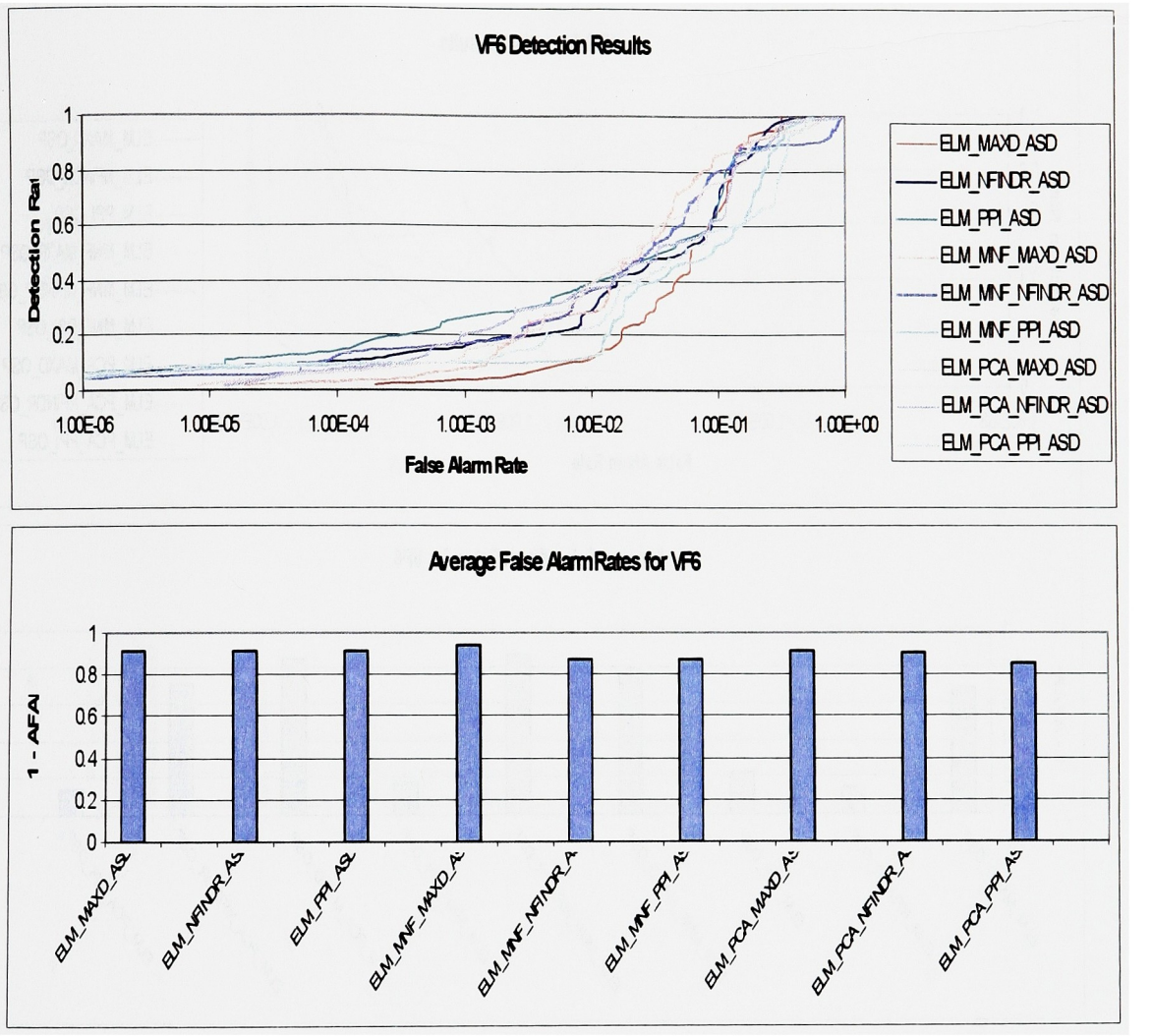

Figure A.45: ROC and (1 – AFAR) graphs for the VF6 target that employed the ELM atmospheric compensation algorithm and the ASD matched filter.

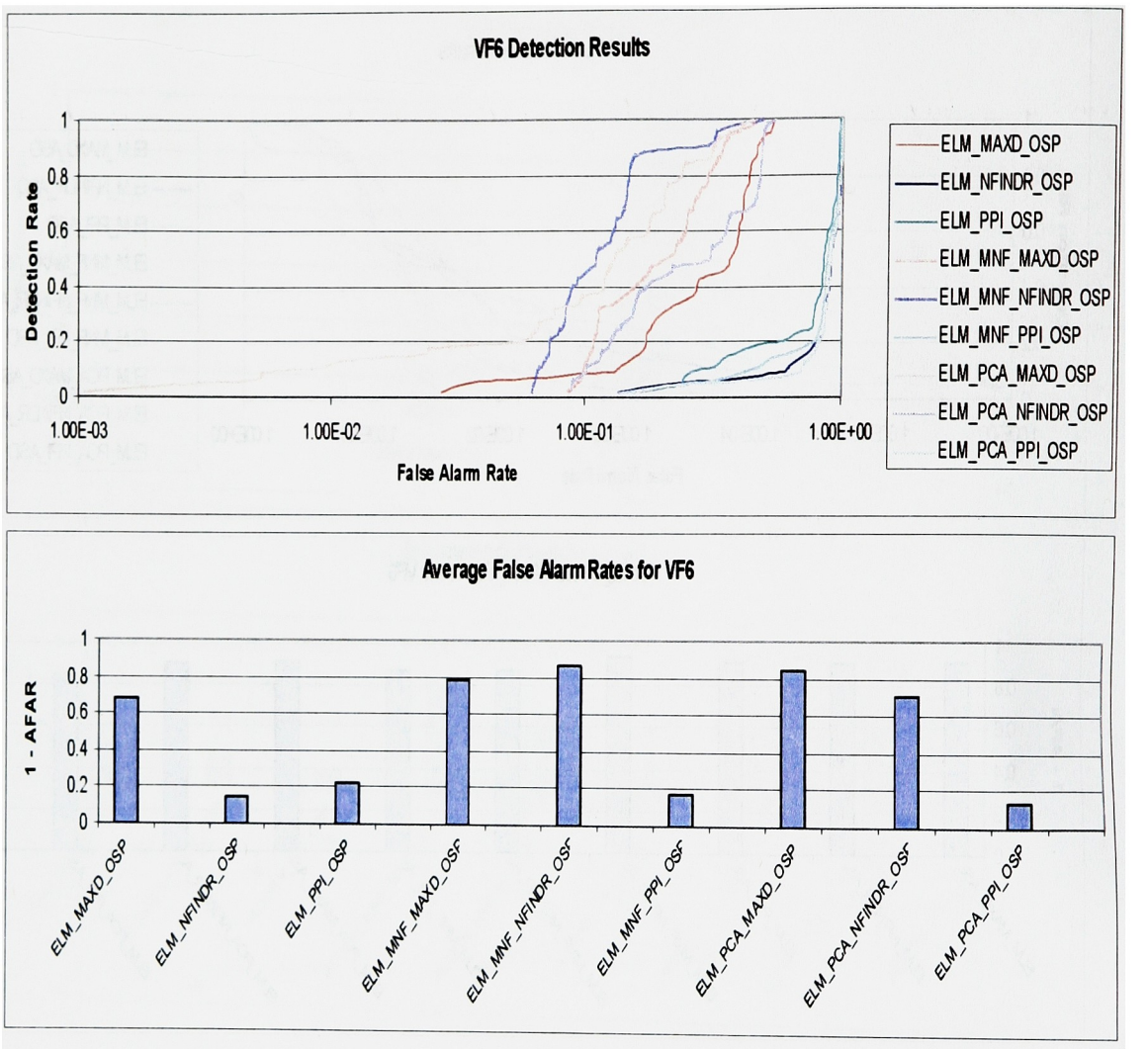

Figure A.46: ROC and (1 − AFAR) graphs for the VF6 target that employed the ELM atmospheric compensation algorithm and the OSP matched filter.

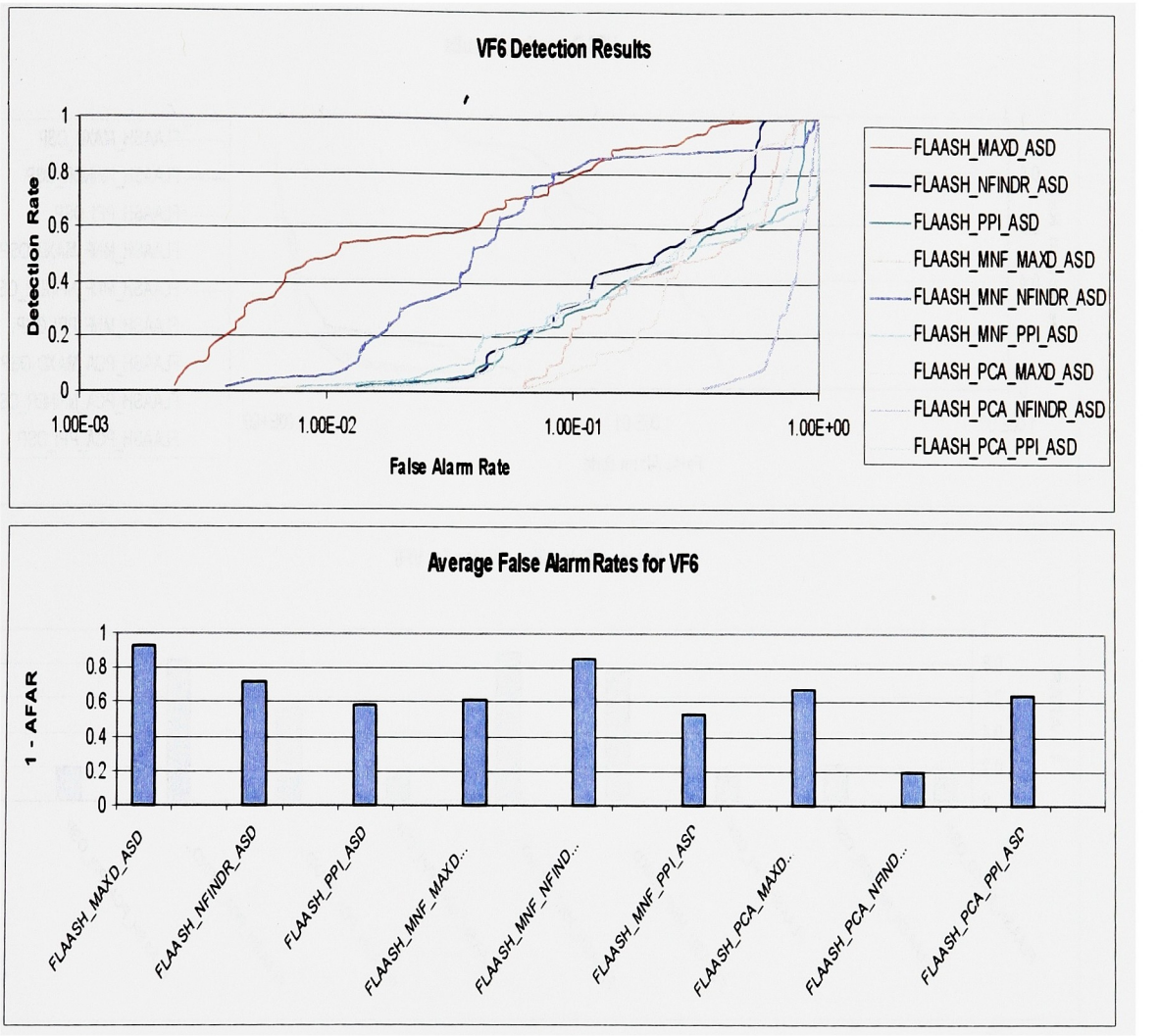

Figure A.47: ROC and (1 – AFAR) graphs for the VF6 target that employed the FLAASH atmospheric compensation algorithm and the ASD matched filter.

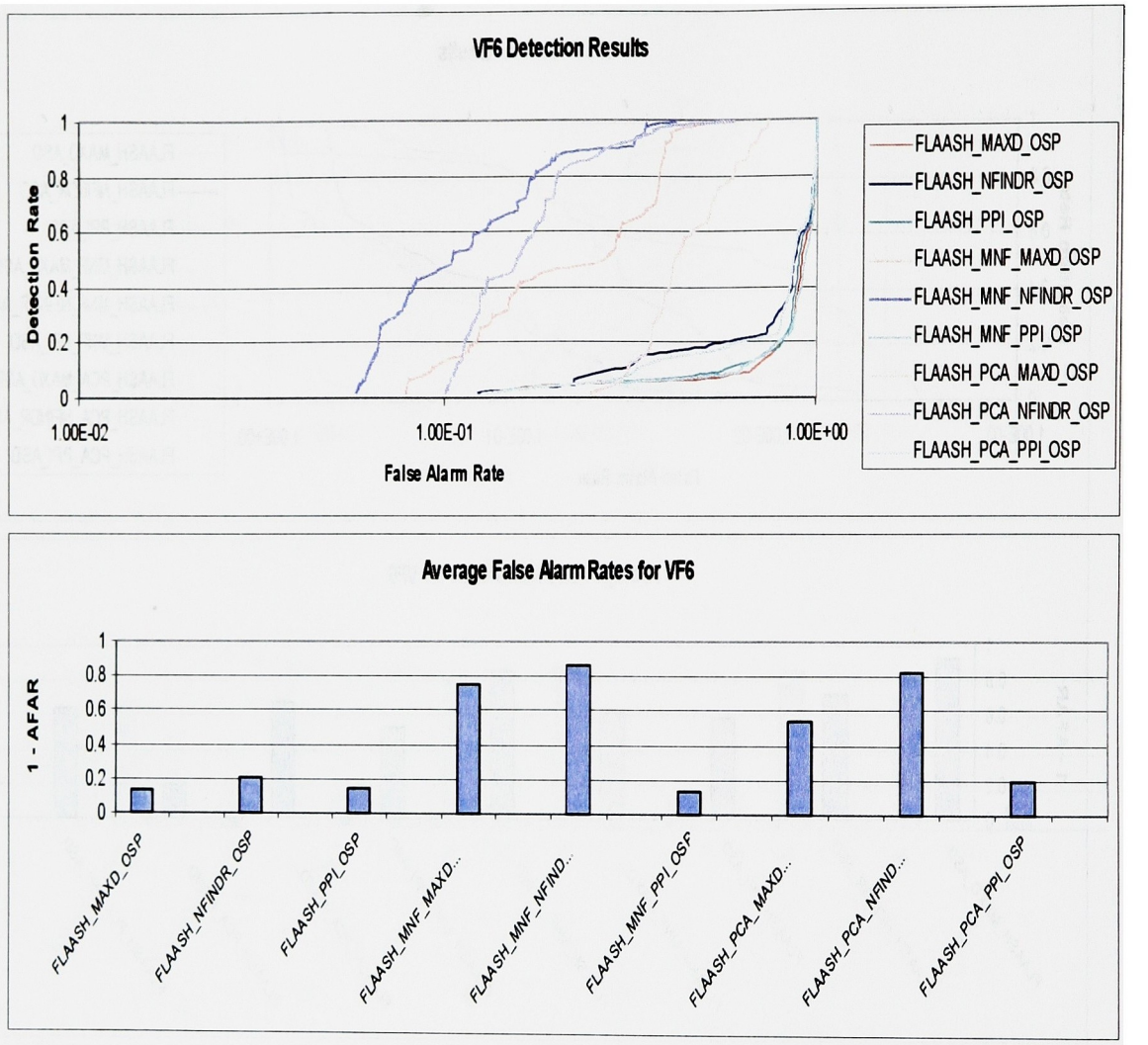

Figure A.48: ROC and (1 – AFAR) graphs for the VF6 target that employed the FLAASH atmospheric compensation algorithm and the OSP matched filter.

Missing Page

Bibliography

Peter Bajorski and Emmett J. Ientilucci. Geometric basis-vector selection methods and subpixel target detection as applied to hyperspectral imagery. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, September 2004.

Peter Bajorski, Emmett J. Ientilucci, and John R. Schott. Comparison of basis-vector selection methods for target and background subspaces as applied to subpixel target detection. In *Proceedings of SPIE*, volume 5245, pages 97–108, August 2004.

A. Berk, S. M. Adler-Golden, A. J. Ratkowski, G. W. Felde, G. P. Anderson, M. L. Hoke, T. Cooley, J. H. Chewynd, J. A. Gardner, M. W. Matthew, L. S. Bernstein, P. K. Acharya, D. Miller, and P. Lewis. Exploiting MODTRAN radiation transport for atmospheric correction: The FLAASH algorithm. In *Proceedings of the Fifth International Conference on Information Fusion*, volume 2, July 2002.

Alexander Berk, Gail P. Anderson, Prabhat K. Acharya, Lawrence S. Bernstein, Leonid Muratov, Jamine Lee, Marsha J. Fox, Steven M. Adler-Golden, James H. Chetwynd, Michael L. Hoke, Ronald B. Lockwood, Thomas W. Cooley, and James A. Gardner. MODTRAN5: A reformulated atmospheric band model with auxiliary species and practical multiple scattering options. In *Proceedings of SPIE*, volume 5655, pages 88–95, January 2005.

Joseph W. Boardman, Fred A. Kruse, and Robert O. Green. Mapping target signatures

via partial unmixing of AVIRIS data. In *Fifth JPL Airborne Earth Science Workshop*, volume 1 of *JPL Publication 95-1*, pages 23–26, 1995.

Andrew A. Green, Mark Berman, Paul Switzer, and Maurice D. Craig. A transformation for ordering multispectral data in terms of image quality with implications for noise removal. In *IEEE Transactions or Geoscience and Remote Sensing*, volume 26, pages 65–74, January 1988.

Robert O. Green, James E. Conel, and Dar A. Roberts. Estimation of aerosol optical depth, pressure elevation, water vapor and calculation of apparent surface reflectance from radiance measured by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) using a radiative transfer code. In *Proceedings of SPIE*, volume 1937, pages 2–11, 1993.

Joseph C Harsayni and Chein-I Chang. Hyperspectral image classification and dimensionality reduction: An orthogonal subspace projection approach. In *Geoscience and Remote Sensing, IEEE Transactions on*, volume 32, issue 4, pages 779–785, July 1994.

ImSpec LLC. *ACORN 5.0 User's Manual.* ImSpec LLC, 2004. Version 040801.

F. A. Kruse. Comparison of ATREM, ACORN, and FLAASH atmospheric corrections using low-altitude AVIRIS data of Boulder, Colorado. In *13th JPL Airborne Geoscience Workshop*, 31 March - 2 April 2004.

Kyungsuk Lee. *A Subpixel Target Detection Algorithm for Hyperspectral Imagery.* PhD thesis, Rochester Institute of Technology, May 2004.

Dimitris Manolakis and Gary Shaw. Detection algorithms for hyperspectral imaging applications. In *IEEE Signal Processing Magazine*, pages 29–43, January 2002.

Dimitris Manolakis, Christina Siracusa, David Marden, and Gary Shaw. Hyperspectral adaptive matched filter detectors: Practical performance comparison. In *Proceedings of SPIE*, volume 4381, pages 18–33, August 2001.

RSI. *FLAASH User's Guide.* Research Systems Incorporated, September 2004. ENVI FLAASH Version 4.1.

Lee J. Rickard, Robert W. Basedow, Edward F. Zalewski, Peter R. Silverglate, and Mark Landers. HYDICE: An airborne system for hyperspectral imaging. In *Proceedings of SPIE*, volume 1937, pages 173–179, September 1993.

John R. Schott. *Remote Sensing: The Imaging Chain Approach.* Oxford University Press, 1997.

Senna Consulting. *N-FINDR 3.0 Documentation.* Senna Consulting, 2004. Distributed with N-FINDR 3.0 software package.

Michael E. Winter. N-FINDR: An algorithm for fast autonomous spectral end-member determination in hyperspectral data. In *SPIE Conference on Imagnig Spectrometry V*, volume 3753, July 1999.